Google Summer of Code

SHAKTI

BeagleWire Software

FPGA Based Verification of SHAKTI Processor

Student : Omkar Bhilare
Mentors: Michael Welling
Lavanya J, prof V. kamakoti

SHAKTI

beagleboard.org

HELLO!

Name : Omkar Bhilare

College : Final Year Electronics Undergraduate Student at VJTI, India

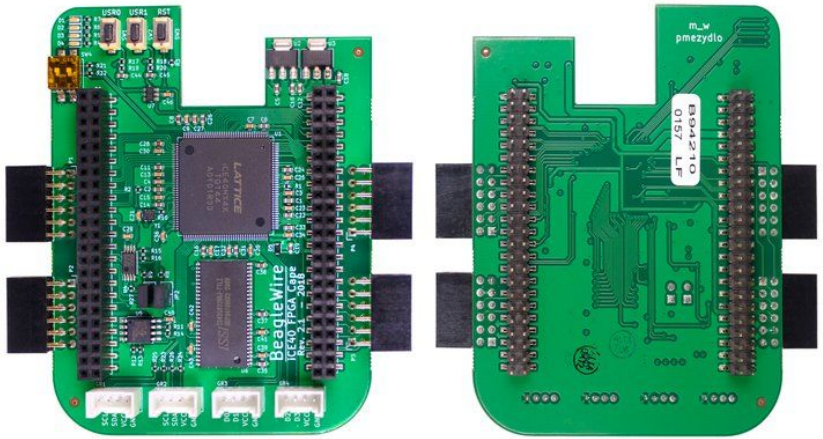
Hardware: FPGAs (Cyclone 2, Tang Primer, ICE40)
MCUs (ESP32/ESP8266, Arduino)

Software : Quartus Prime, Modelsim, Icarus Verilog, gtkwave, Autodesk Eagle, Altium, Kicad, Proteus, Multisim, Logisim, Git, Linux

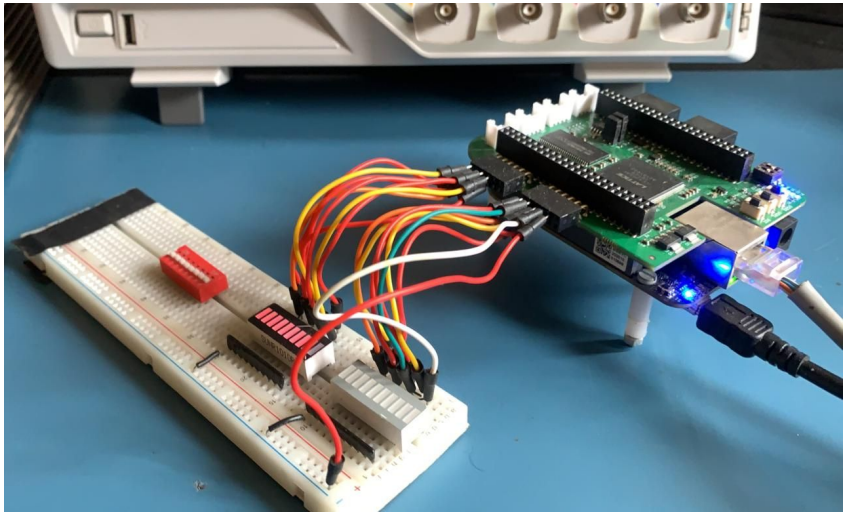
Interest : FPGA, ASICs, Digital VLSI, Electronics.

oabhilare b19@el.vjti.ac.in github.com/ombhilare999 linkedin.com/omkar-bhilare

BeagleWire



BeagleWire



My Setup

BEAGLEWIRE

Verilog IPs:

- GPMC to Wishbone

Read to Use Examples:

- Arm Blink LEDs
- Blink LEDs
- Bar Graph with Wishbone Intercon
- VGA Test Screens
- VGA PONG GAME
- Encoder Example

Programming Methods:

- ICE40-spi kernel Method
- SPI Flash Method

Litex Support:

- Litex Core with:
  - Award winning Serv
  - LITEDRAM
  - UART-Wishbone Crossover

Protocols Support:

- External IPs with wishbone port can be easily interfaced with Beaglewire

PMOD Support:

- VGA PMOD
- Encoder PMOD
- LED PMOD
- Switches PMOD

Increase Documentation:

- Starting Guide
- Each Example-Driver Guide
- Beaglewire Litex Guide
- beaglewire.github.io

Programming BeagleWire

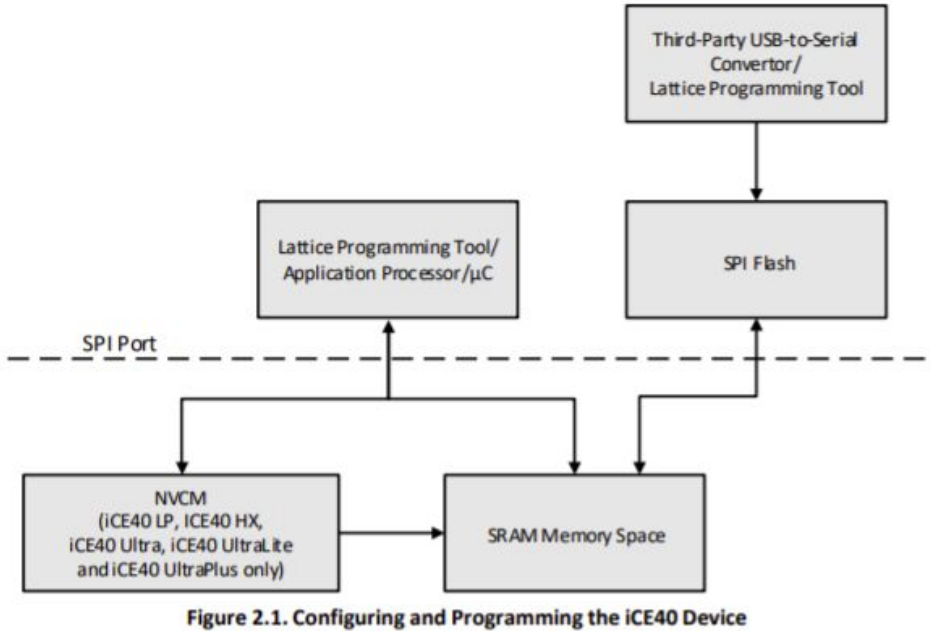
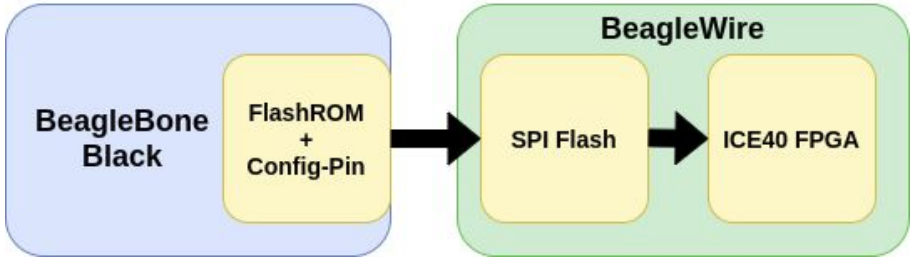


Figure 2.1. Configuring and Programming the iCE40 Device

- **Programming Methods:**
  - **ICE40-spi kernel Method**
    - **Overlay Name:** BW_ICE40Cape_00A0_LKM.dtbo
  - **SPI Flash Method**
    - **Overlay Name:** BW_ICE40Cape_00A0.dtbo



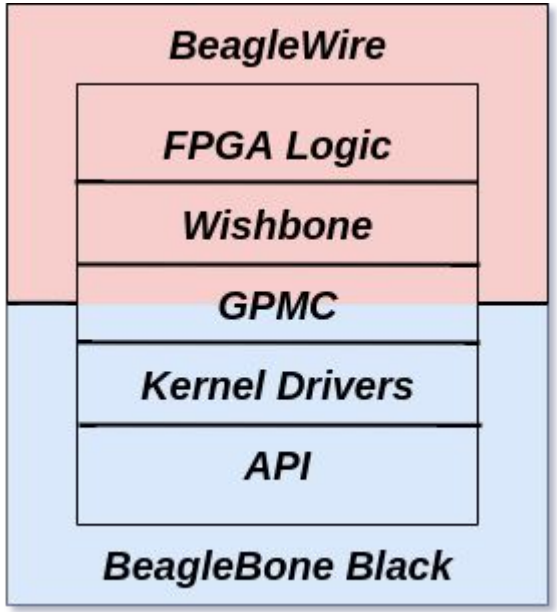
Communication Between BeagleWire and BBB

WISHBONE COMPATIBLE

BEAGLEWIRE

Advantages of GPMC:

- GPMC_AD is a 16-bit bus responsible for transfer address and data from/and to the BeagleBone, making it an input and output bus with multiple purposes.
- The use of this bus inside the FPGA to interact with the blocks not only complicates the whole project but also enforces the use of more logic occupying more space inside the FPGA.
- Wishbone comes from the need of creating a common interface between IP cores improving the portability of projects and reliability.
- It separates the data written, data read and address in distinctive buses, making it more easy and straightforward to use. For that reason was chose to decode the GPMC into wishbone signals.

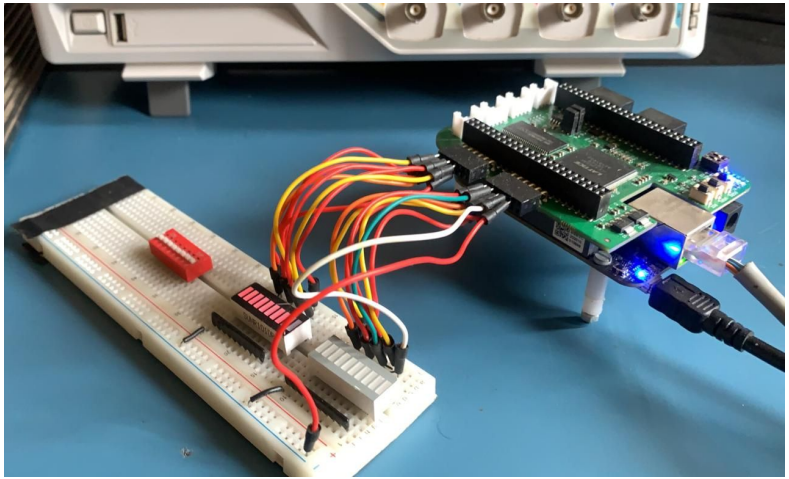


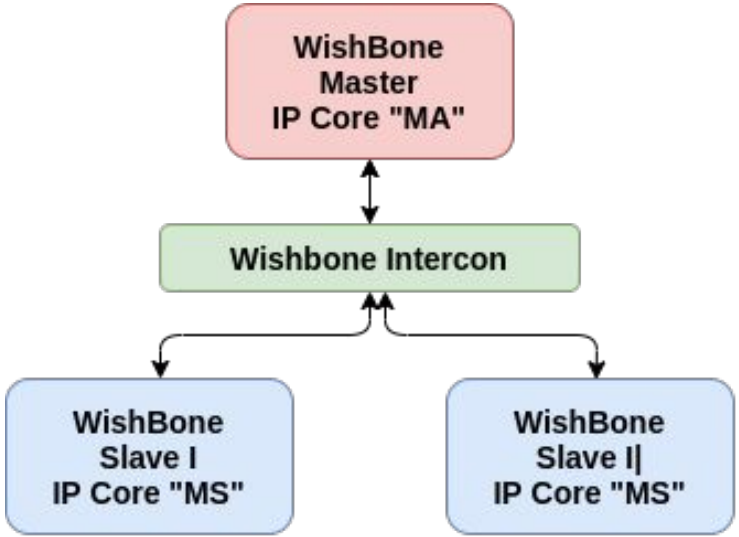
Wishbone Examples with Intercon Support

WISHBONE COMPATIBLE

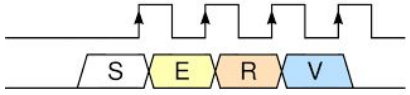
BEAGLEWIRE

- **Read to Use Wishbone Examples:**
  - Arm Blink LEDs
  - Bar Graph with Wishbone Intercon





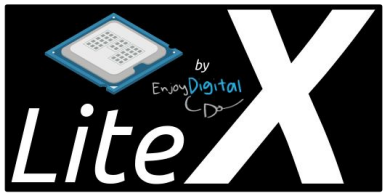
Litex Supports on BeagleWire



+

+

Wishbone over UART

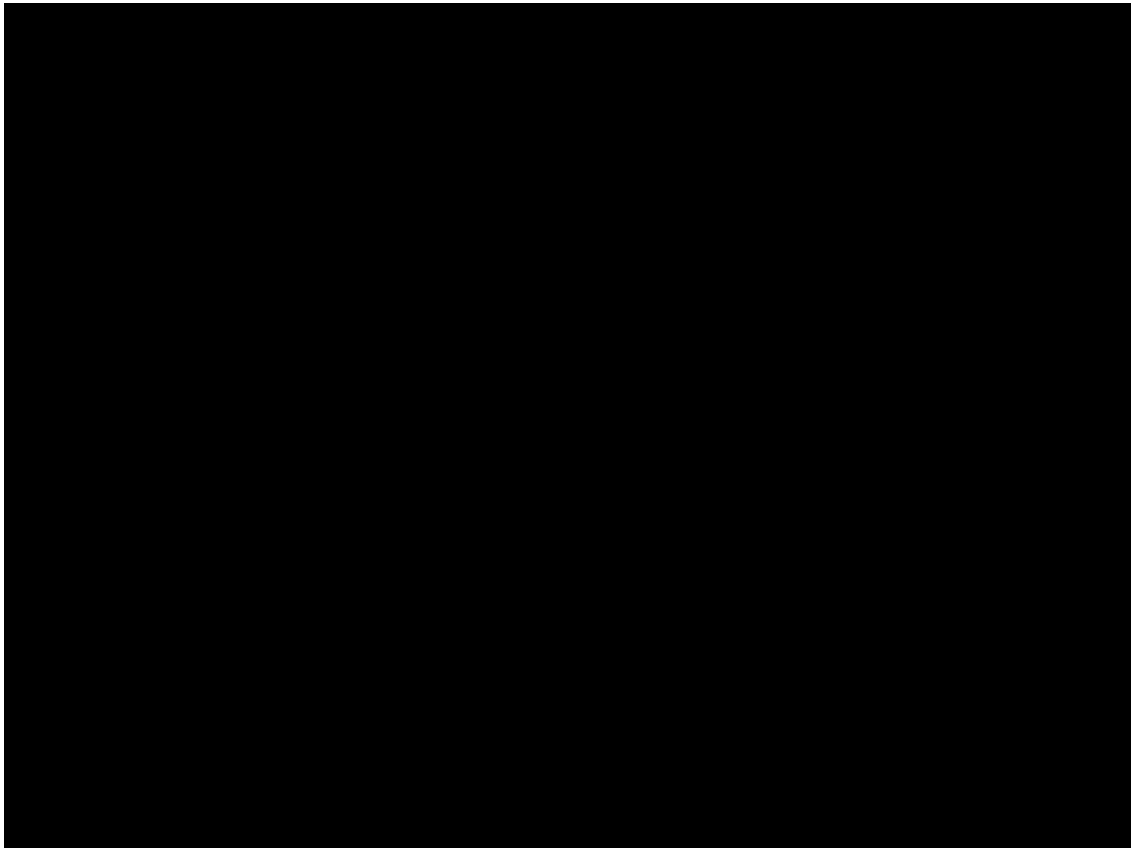


Build your hardware, easily!

PMOD Support of BeagleWire:

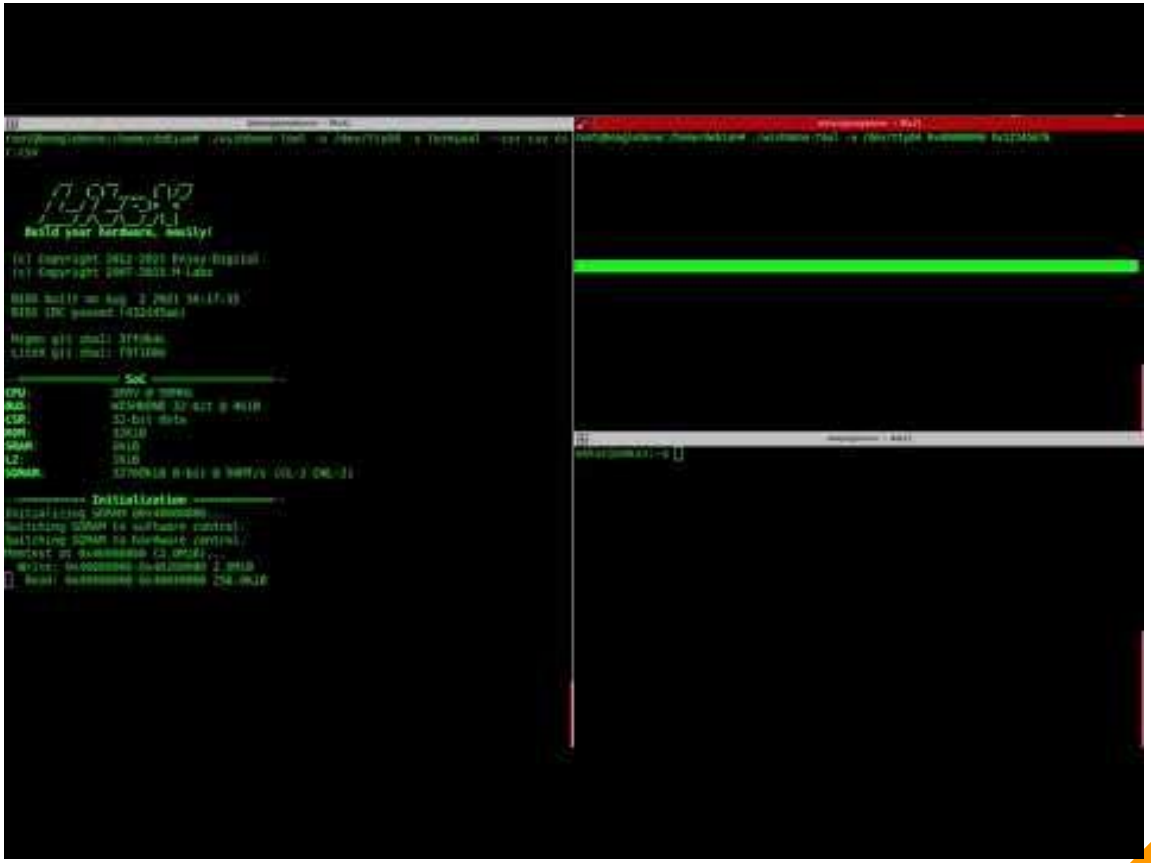
VGA PMOD	Encoder PMOD	LED/Bar-Graph PMOD	Switch PMOD
			

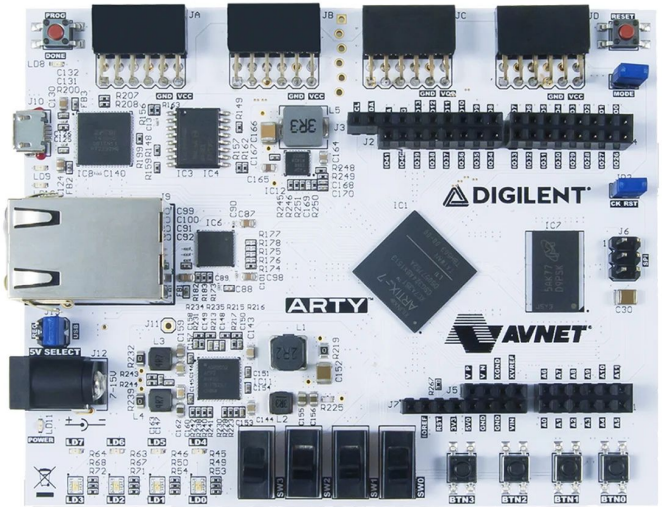
PONG on VGA DEMO:



SPI Flash DEMO:

Litex-BeagleWire DEMO:





Artix-7 FPGA

VAJRA

A 64-bit C-class SoC on Artix7 100T board

Features:

- C-class 6 stage In-order core
- RV64IMACSU supported
- 256 MB of DDR and 4 KB of ROM
- Arduino Compatible pin assignments
- Pin Mux supported
- 32 GPIOs
- 6 PWM controllers
- 2 SPI controllers
- 3 UART controllers
- 2 I2C controllers
- 1 Timer (CLINT)
- One Interrupt controller (PLIC)
- On board XADC
- Onboard FTDI based debugger
- Physical Memory Protection enabled
- Soft Float library support
- Ethernet Lite

VAJRA SOC

Project Report

FPGA Based C-Class Verification

Omkar Bhilare
omkarbhilare45@gmail.com
Mentor: Lavanya J
lavanya.jagan@gmail.com
Shakti Project, IIT Madras

1 Abstract:

C-Class is a member of the SHAKTI family of processors. It is an extremely configurable and commercial-grade 5-stage in-order core supporting the standard RV64GCSUN ISA extensions. Verification is a very important process in chip design. Usually, the software RTL simulation tests on host computers take a very large amount of time. This paper explores the possibility of FPGA-based verification of SHAKTI processors to reduce verification time. AAPG is a tool that is intended to generate random RISC-V programs to test RISC-V cores. In this paper, we ran AAPG tests on FPGA and compared the spike's golden signature dump with the FPGA signature dump. This paper also explores the self checking tests generated by AAPG. Self-checking tests have the advantage of running on FPGA or silicon without much intervention from the host, thereby accelerating the speed verification significantly.

Key Words: C-Class, FPGA, RTL, AAPG, RV64GCSUN, ISA, Verification

FPGA Based Verification of SHAKTI Processors

Comparision of RTL Verification Approach			
RTL Verification Approach	**Speed**	**Controllability**	**Visiblity**
Software RTL Simulation	Very Slow	High	Full
Hardware Emulation Engines	Very Fast	High	Full
FPGA based Prototyping	Fast	Low	Limited

Table 1: Comparision of RTL Verification Approach

SHAKTI FPGA Framework

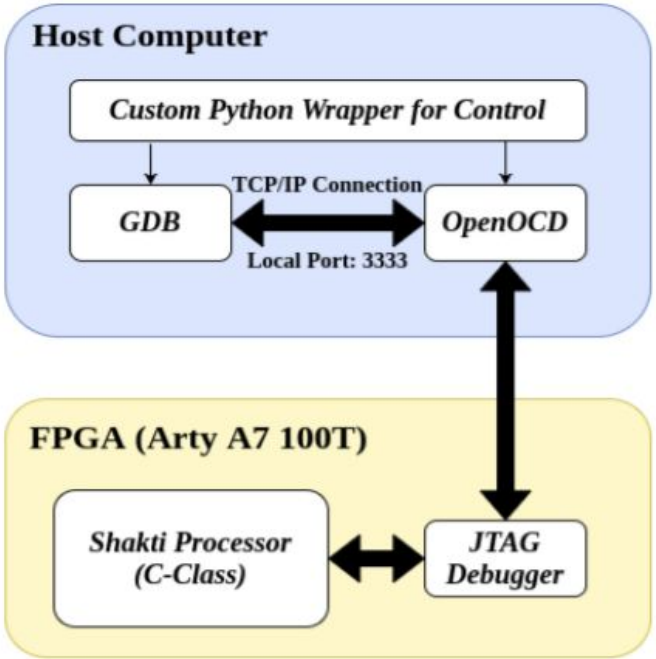
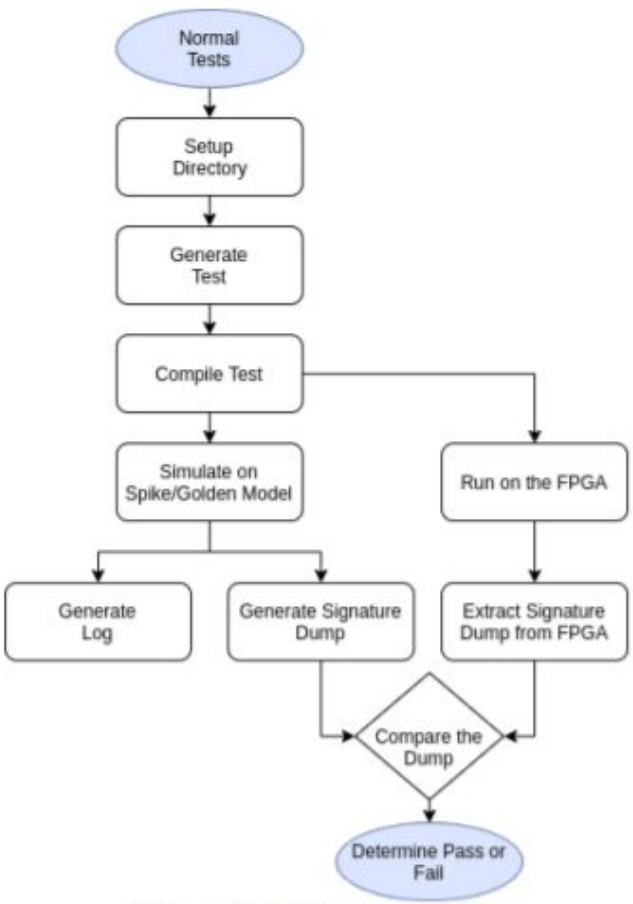
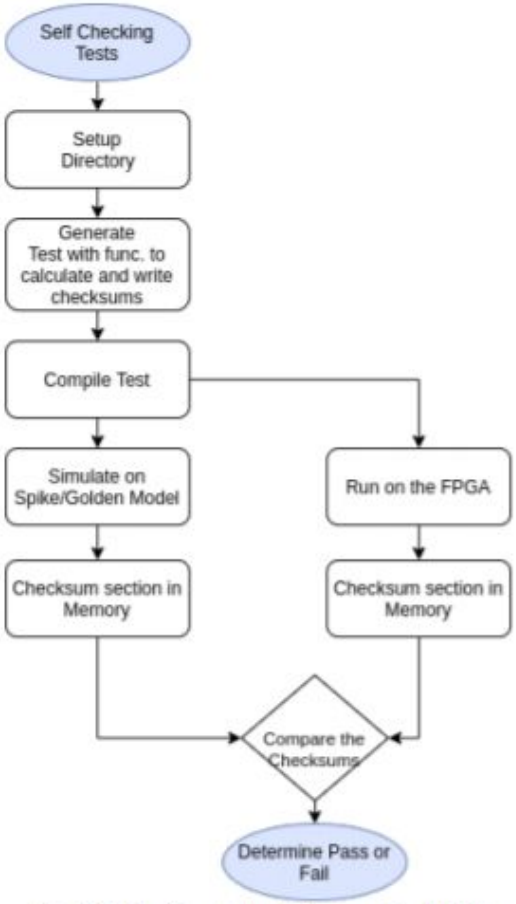


Figure 1: Shakti FPGA framework



(a) Normal AAPG Tests on the FPGA

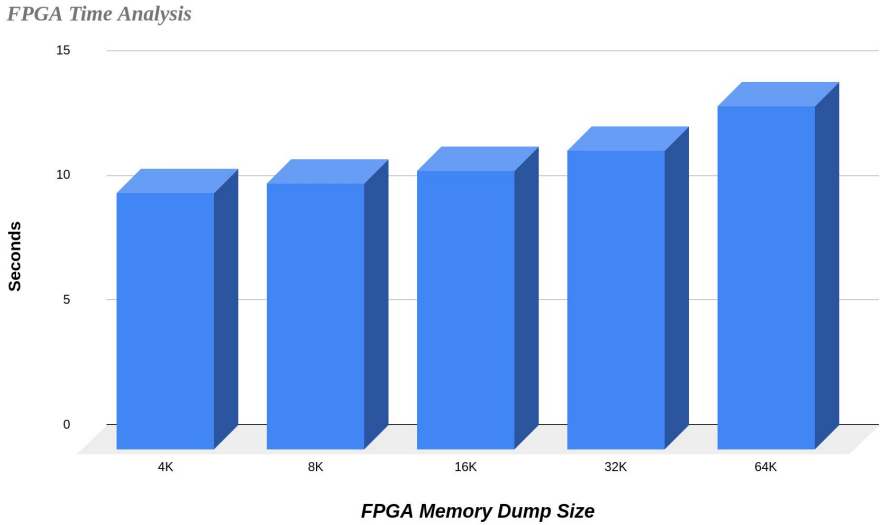


(b) Self-Checking AAPG Tests on the FPGA

Figure 2: FPGA Test Flow

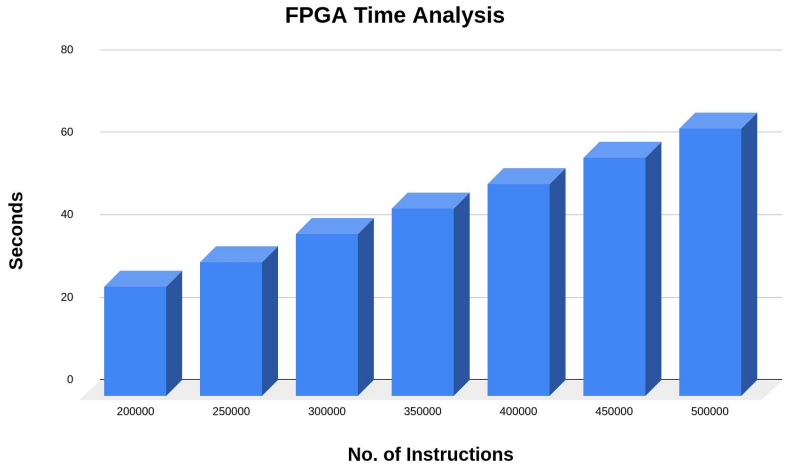
Memory Dump Test:

AAPG Normal TESTS					
Test Sr No	ISA Distribution	No. of Instructions	Dump Size	Dump Range	Time (Seconds)
1	rel_rv64i.compute: 1 rel_rv64i.data:1 rel_rv64a:1 rel_rv64m: 1	80000	4K	0x80091000 :: 0x80092000	10.261164
2			8K	0x80091000 :: 0x80093000	10.646166
3			16K	0x80091000 :: 0x80095000	11.153076
4			32K	0x80091000 :: 0x80099000	11.968721
5			64K	0x80091000 :: 0x800A0FFF	13.724016



Instruction Test:

AAPG SELF CHECKING TESTS					
Test Sr No	ISA Distribution	No. of Instructions	Dump Size	Dump Range	Time (Seconds)
1	rel_rv64i.compute: 1 rel_rv64i.data:1 rel_rv64a:1 rel_rv64m: 1	200000	16K	0x80a01000 :: 0x80a05000	26.429611
2		250000	16K	0x80a01000 :: 0x80a05000	32.305991
3		300000	16K	0x80a01000 :: 0x80a05000	39.114424
4		350000	16K	0x80a01000 :: 0x80a05000	45.324175
5		400000	16K	0x80a01000 :: 0x80a05000	51.337632
6		450000	16K	0x80a01000 :: 0x80a05000	57.779498
7		500000	16K	0x80a01000 :: 0x80a05000	64.681013



Results:

These tests were ran on the Shakti's VAJRA SOC. The Major Advantage of FPGA in verification flow is the Speed. As shown in the Figure 3, 5,00,000 RISC-V Instruction took only 64 Seconds to run on the FPGA. The 5 visibility and control is required in any Verification Framework. Traditional FPGA flow has usually less visibility and control but with the help of checksums in AAPG and python wrapper we can see where is the mismatch. In future we need to run this framework on the Shakti's I-class.

GSoC Certificate:

SHAKTI Letter:

INDIAN INSTITUTE OF TECHNOLOGY, MADRAS
Department of Computer Science and Engineering
Chennai 600 036, India
Tel: 2257 4368
Fax: 2257 4352
Email : kama@cse.iitm.ac.in

Dr. V. Kamakoti
Professor

Date: 06.08.2021

TO WHOMSOEVER THIS MAY CONCERN

This is to certify that **Mr. Omkar Bhilare (Roll No: 191060901)**, B. E Department of Electrical Engineering, **Veermata Jijabai Technological Institute, Mumbai** has worked in the project titled **"FPGA Based Verification of SHAKTI Processors"** at Reconfigurable and Intelligent Systems Engineering (RISE) lab, Department of Computer Science and Engineering, IIT Madras as a part of his internship during **11th March 2021 to 31st July 2021.**

V. Kamakoti

Prof. V. Kamakoti

Prof. V. KAMAKOTI * Professor of Computer Science and Engineering *

THANKS!

Any questions?

You can find me at

Twitter: @ombhilare999 & oabhilare_b19@el.vjti.ac.in