Shakti AARDONYX SoC Verification

FPGA Emulation of Aardonyx SoC - Specification

FPGA	-	**Arty A7 100t**
Core	-	**Shakti E-Class**
ISA	-	**RV32IMAC**
Core Clock	-	**50MHz**

Branch Link
https://gitlab.com/shaktiproject/cores/shakti-soc/tree/master/asic/e-class-aardonyx/fpga/test

As we don't have an FMC connector for connecting SDRAM, we decided to use BRAM as memory.

BRAM	-	**512 Kb**

The default UART baud rate is configured as 9600.

Repository Overview:

Master branch: shakti-soc/asic/e-class-aardonyx

Branch created: `32-adding-verification-repository-for-e-class-aardonyx`

Branch description:

fpga/test/ → artya7-100t (software tests)
→ artya7-100t-eval (FPGA Emulation of Aardonyx SoC)

artya7-100t-eval→ Tcl files to generate the bitstream for aardonyx soc on arty 100t

artya7-100t → This directory has all the software test codes for validation of all the functionality of the peripherals.

→ periperal-tests (contains all the tests for peripherals)
→ Makefile (to compile the codes with respect to the Aardonyx board)

periperal-tests

→ **gpio** (General Purpose Input Output)

→ **rdgpio** - to read the gpio pins
→ **tglgpio** - to write to the gpio pins

→ **i2c (**Inter-integrated Circuit)

→ **lm75** - to read the temperature values from sensor

→ **pwm** (Pulse Width Modulation)

→ **pwmled** - motor speed control by varying the duty cycle (Ton & Toff)

→ **qspi** (Quad Serial Peripheral Interface)

→ **issi_flash** - single, double, quad mode & DDR XIP

→ **spi** (Serial Peripheral Interface)

→ **spansion_flash** - flash erase, write, read

→ **uart** (Universal Asynchronous Receiver/Transmitter)

→ **hello** - print data on uart console
→ **uart1** - read and print data

Test cases to verify the functionality of all peripherals on Arty A7 35t FPGA:

1. **GPIO x 16**
   - rdgpio - set all gpio pins as Input and check whether you can read it as HIGH or LOW
   - tglgpio - set all gpio pins as Output and check if it is able to write HIGH or LOW.

The above test cases has been done and tested with a relay as OUTPUT and Moisture sensor as INPUT on the GPIO pins. So the functionality of read and write has been tested.

2. **I2C x 2**
   - lm75 - read the temperature value from the sensor.

3. **SPI x 3**
   - s25fl128s - read and write values into flash memory (cypress flash)

4. **UART x 3**
   - hello - print characters, integer, decimal, hexadecimal on uart console
   - uart1 - read and write values on uart1 & uart2

5. **PWM x 6**
   - motor - varying the duty cycle (Ton and Toff) to control the speed of the dc motor
   - led - varying the duty cycle to control the brightness of the led

6. **QSPI x 1**
   - issi flash - read and write values into flash memory (checked single, double. Quad modes and XIP)

7. **PINMUX x 11**

PINMUX 1 - GPIO0 / UART1_RX
PINMUX 2 - GPIO1 / UART1_TX
PINMUX 3 - GPIO2 / UART2_RX
PINMUX 4 - GPIO3 / UART2_TX / PWM0
PINMUX 5 - GPIO5 / PWM1
PINMUX 6 - GPIO6 / PWM2
PINMUX 7 - GPIO9 / PWM3
PINMUX 8 - GPIO10 / SPI2_NCS / PWM4
PINMUX 9 - GPIO11 / SPI2_MOSI / PWM5
PINMUX 10 - GPIO12 / SPI2_MISO
PINMUX 11 - GPIO13 / SPI2_CLK

For configuring a particular pins behaviour, we need to set the pinmux config register accordingly. The Pinmux_Config register is a 32 bit register and each pin is configured by 2 bits (00, 01, 10).

Pin Configuration with respect to PINMUX:

1. **GPIO**

   GPIO0 - PINMUX 1
   GPIO1 - PINMUX 2
   GPIO2 - PINMUX 3
   GPIO3 - PINMUX 4
   GPIO4
   GPIO5 - PINMUX 5
   GPIO6 - PINMUX 6
   GPIO7
   GPIO8
   GPIO9 - PINMUX 7
   GPIO10 - PINMUX 8
   GPIO11 - PINMUX 9
   GPIO12 - PINMUX 10
   GPIO13 - PINMUX 11
   GPIO14
   GPIO15

For using all GPIOs, set Pinmux Config Register as 0x00000000.

2. **UART**

   UART0 - 11300 (default) (used for system prints)

   UART1 - 11400 (PINMUX_CONFIG value = 0x5)
   PINMUX 1 - UART1_RX
   PINMUX 2 - UART1_TX

   UART2 - 11500 (PINMUX_CONFIG value = 0x50)
   PINMUX 3 - UART2_RX
   PINMUX 4 - UART2_TX

3. **SPI**

   PINMUX 8 - SPI2_NCS
   PINMUX 9 - SPI2_MOSI
   PINMUX 10 - SPI2_MISO
   PINMUX 11 - SPI2_CLK

For using SPI2, set Pinmux Config Register as 0x154000.

4. **PWM**

   PINMUX 4 - PWM0 (Pinmux Config Register = 0x80)

PINMUX 5 - PWM1 (Pinmux Config Register = 0x200)
PINMUX 6 - PWM2 (Pinmux Config Register = 0x800)
PINMUX 7 - PWM3 (Pinmux Config Register = 0x2000)
PINMUX 8 - PWM4 (Pinmux Config Register = 0x8000)
PINMUX 9 - PWM5 (Pinmux Config Register = 0x20000)

For using all the pins as PWM, set Pinmux Config Register as 0x2AA80.