RISC-V SoC and Firmware Development

Lab: 05

Objective

- Get fimilarize with firmware.
- Install and configure the RISC-V toolchain on local machines to compile and run RISC-V programs.
- Convert C programs into assembly, machine code,suitable for RISC-V processors.

Firmware

- Specialized software embedded in device's hardware to control their functions.
- It acts as the intermediary between the hardware and software.
- The purpose of firmware is to configure the instructions for peripherals without changing hardware.

RISC-V toolchain

- The RISC-V toolchain(GNU) is a set of software tools used to develop, compile, and debug software applications for the RISC-V instruction set architecture (ISA),which work together to translate source code into machine code that can be executed on a RISC-V processor.
- **It includes,**
- **Compiler**: translates source code into assembly code
- **Assembler**: translates assembly code into machine code
- **Linker**: combines multiple object files into a single executable file
- **Debugger**: allows developers to test and debug their code

Getting Started with RISC-V Toolchain

- The purpose of using RISC-V toolchain in this project is to optimze firmware for our hardware.

Installation & Build

STEP 01:

- Follow the URL(https://github.com/stnolting/riscv-gcc-prebuilt) and download the second option.

Status	Release (tag)	Download	GCC	binutils	march	mabi	c-lib
🟢	rv32e-231223	download	13.2.0	2.41	`rv32e`	`ilp32e`	newlib
🟢	rv32i-131023	download	13.2.0	2.41	`rv32i`	`ilp32`	newlib
🔴	rv32i-4.0.0	download	12.1.0	2.39	`rv32i`	`ilp32`	newlib

Getting Started with RISC-V Toolchain

STEP 02:

- Create a folder where you want to install the toolchain an type
- **$ sudo mkdir /opt/riscv**

STEP 03:

- Navigate to the download folder, **/opt/riscv** is the path where you want to install toolchain.
- **$ sudo tar -xzf TOOLCHAIN.tar.gz -C /opt/riscv/**
- **$ export PATH=$PATH:/opt/riscv/bin**

STEP 04:

- **$ riscv32-unknown-elf-gcc -v**

Converting C to Hex

- **STEP 01:**
- Make a file in C for adding two numbers and print

```c
#include <stdio.h>

int main() {
    int a = 10;
    int b = 20;
    int result = a + b;
    printf("The result is: %d\n", result);
    return 0;
}

```

Open the terminal and run the following command, which will generate and object file for the C file.

$ riscv32-unknown-elf-gcc -o add add.c

Converting C to Hex

STEP 02:

$ riscv32-unknown-elf-objcopy -O ihex add add.hex

This command will generate the hex file from the executable file generated in previous step.

Optional step:

$ riscv32-unknown-elf-objdump -d add > add.asm

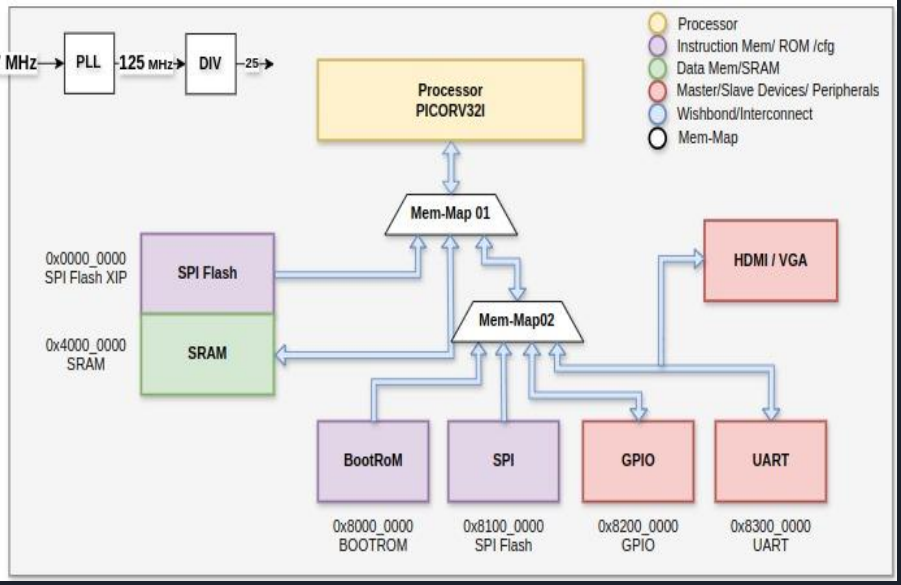
- **-d:** Specifies to disassemble the machine code into assembly instructions.
- **add:** is the executable file

Converting C to Hex

```
typedef struct {
    volatile uint32_t OUT;
} PICOGPIO;

define GPIO0 ((PICOGPIO*)0x82000000)
GPIO0->OUT ^= 0x00000004;
```

```
lui     a4,0x82000
lw      a3,0(a4)
xori    a3,a3,4
sw      a3,0(a4)
```



TESTIMONIAL

Author:

Abdul Muheet Ghani, Research Associate at MERL-UITU.

Under The Supervision Of:

Dr.Ali Ahmed (Team Lead MERL).

Sponsored By: Edmund from Symbiotic EDA. for sponsoring FPGA.

Thanks: Lushay Lab (They've provided crucial resources and guidance throughout the project)