Summer Project

B Sathiya Naraayanan
IMT2020534

ISA

RISC-V

- RISC-V is an open standard ISA that is not owned or controlled by any company or organization, but rather by a global community of researchers, developers, and users who collaborate to create and improve the ISA.
- RISC-V is designed to be simple and elegant, avoiding unnecessary complexity and redundancy.
- RISC-V uses fewer instructions than other ISAs, which reduces the size and power consumption of the microchip.
- RISC-V also enables better optimization and parallelization of the microchip operations, which improves the performance and throughput of the microchip.
- RISC-V is compatible with various hardware platforms and software ecosystems, thanks to its open and standard nature.

ARM

- ARM is a proprietary ISA that is owned and licensed by ARM Holdings, a subsidiary of Nvidia. ARM is fixed and standardized, meaning it has a predefined set of instructions
- ARM uses more instructions than RISC-V, which increases the size and power consumption of the microchip.
- ARM also executes multiple instructions per clock cycle, which improves the performance and throughput of the microchip.
- ARM is compatible with various hardware platforms and software ecosystems, thanks to its dominant and popular position in the microchip industry.
- ARM runs on most types of microchips, such as CPUs, GPUs, FPGAs, ASICs, and SoCs.

Requirements for IOT

Requirements for IOT

In most typical IoT applications we can find the following processor system requirements:

- Low power
- High code density
- High performance
- Memory space and addressing modes
- OS friendly
- Security

Low Power

- **Sleep Modes:** Architectural sleep modes, clock gating, and multi-power domains to maintain low active power and ultra-low idle power.
- **Energy Efficiency:** Energy-efficient design is crucial for battery-powered products and reduces overall energy consumption in always-on systems.
- **Reduced EMI:** Lower power consumption reduces Electro Magnetic Interference (EMI), beneficial for wireless and safety-critical applications, enhancing communication quality and reducing interference.

High Code Density

1. **Thumb-2 Technology:** Thumb-2 instruction set is used in cortex, which combines 16-bit and 32-bit instructions for better code density.
2. **Memory Efficiency:** High code density allows for smaller flash memory usage, reducing the cost and power consumption of IoT products.
3. **Bus Bandwidth:** Efficient code density reduces instruction fetches, freeing up bus bandwidth for other data transfers, which is important in devices with fast Ethernet interfaces.

High Performance

1. **32-bit Architecture:** Processors offer higher performance than 8-bit and 16-bit microcontrollers, ensuring reliability in unpredictable network activity.
2. **Security Capabilities:** High processing power allows for additional security measures, such as encryption and authorization, improving the security of IoT devices.
3. **User Experience:** Better performance supports advanced features like graphical user interfaces and enhances the overall user experience.

Memory Space and Addressing Modes

Large Address Space: Processors support a 32-bit linear address space, allowing up to 4GB of memory, avoiding limitations seen in 8-bit and 16-bit microcontrollers.

C-Friendly Architecture: Various addressing modes make the architecture easy to program and debug, enhancing performance and software development.

Increased Reliability: Larger memory space supports more features and better performance, handling increased internet activity and ensuring system reliability.

OS Friendly

Dual Stack Pointer: Dual stack pointer mechanism for easy OS implementation, efficient memory usage, and reliable operation.

System Tick Timer: A dedicated 24-bit timer for OS System Tick interrupt generation simplifies OS porting across different Cortex-M devices.

MPU Support: The optional Memory Protection Unit (MPU) improves reliability and security by isolating memory access and preventing data corruption.

Security

Privileged Execution Levels: Combining the MPU with privileged and unprivileged execution levels enhances security, limiting the impact of potential vulnerabilities.

Performance for Security: High performance supports sophisticated security measures, such as input validation and encryption, even without hardware accelerators.

Fault Handling: Processors include fault exception handling capabilities to detect and respond to errors, improving the system's ability to recover from attacks.

Comparison

Low Power

PicoRV32:

- Designed for minimal resource usage, which inherently leads to low power consumption.
- Uses a simple, minimalistic architecture with a focus on energy efficiency.

Cortex-M4:

- Optimized for low power consumption with several power-saving modes.
- Includes a sleep mode and deep sleep mode to further reduce power usage when the processor is idle.

High Code Density

PicoRV32:

- Based on the RISC-V architecture, which uses a simple instruction set that may result in lower code density compared to more complex architectures.
- Less efficient in terms of code density compared to ARM's Thumb-2 instruction set.

Cortex-M4:

- Utilizes the Thumb-2 instruction set, which provides a good balance between high performance and high code density.
- Typically achieves higher code density than traditional RISC architectures, making it more memory efficient.

High Performance

PicoRV32:

- A minimalistic design, which limits its performance capabilities compared to more complex processors.
- Suitable for basic control and simple computational tasks but not designed for high-performance applications.

Cortex-M4:

- Features a Harvard architecture with a three-stage pipeline for higher performance.
- Includes DSP instructions and a Floating Point Unit (FPU) for enhanced performance in computationally intensive tasks.

Memory Space and Addressing Modes

PicoRV32:

- Supports a simple memory model with basic addressing modes.
- Typically used in systems with smaller memory requirements due to its simplicity and minimalistic design.

Cortex-M4:

- Supports a 4GB address space with various addressing modes, including direct, indirect, and indexed.
- More sophisticated memory management capabilities, suitable for larger and more complex applications.

OS Friendly

PicoRV32:

- Limited in OS support due to its simple design and lack of hardware features required by more complex operating systems.
- More suitable for bare-metal or simple real-time operating systems (RTOS).

Cortex-M4:

- Well-supported by various RTOS and even lightweight versions of more complex operating systems.
- Includes features such as a SysTick timer and NVIC (Nested Vectored Interrupt Controller) that enhance its suitability for OS-based applications.

Security

PicoRV32:

- Lacks advanced security features found in more complex processors.
- Security must be implemented at the software level, which may not be as robust.

Cortex-M4:

- Supports several security features, including memory protection units (MPUs) and TrustZone for ARMv8-M.
- Provides a more secure environment for IoT applications requiring enhanced security measures.

References

References

- https://www.icdrex.com/risc-v-vs-arm-which-open-standard-risc-architecture-is-better-for-your-microchip-project/#:~:text=However%2C%20some%20of%20the%20advantages,is%20more%20complex%20and%20powerful.
- https://blog.segger.com/code-size-closing-the-gap-between-risc-v-and-arm-for-embedded-applications/
- https://s-o-c.org/cortex-m4f-vs-m4-how-do-these-arm-cores-compare/
- https://xml.jips-k.org/full-text/view?doi=10.3745/JIPS.03.0129
- https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10305479