HABITAT: HARDWARE-ACCELERATED BINARY TRANSLATOR

Sing Hin To (Jason), Tom Spink

School of Computer Science, University of St Andrews

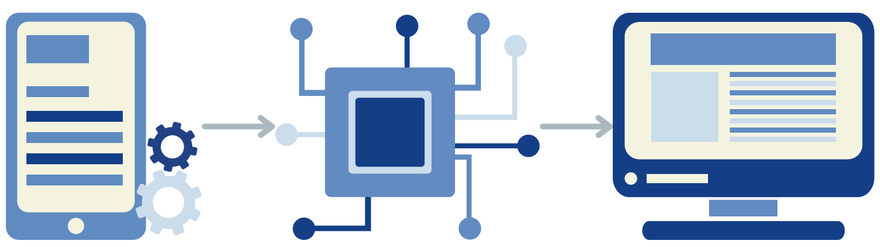
University of St Andrews

1 BACKGROUND

Emulators have been essential in supporting legacy systems, future prototypes, and programs compiled for other architectures.

Intermediate representations (IRs) are often used together to encode details of the foreign system, while emulators **translate** and **execute** programs on the host system based on those IRs.

Examples include QEMU and BlueStacks, which are used in both **commercial** and **recreational** settings.



2 PROBLEM STATEMENT

Software-based solutions often face **performance** and **scalability** issues due to the combined workload on the CPU. This raises the **setup cost** while reducing the **potential** for emulation.

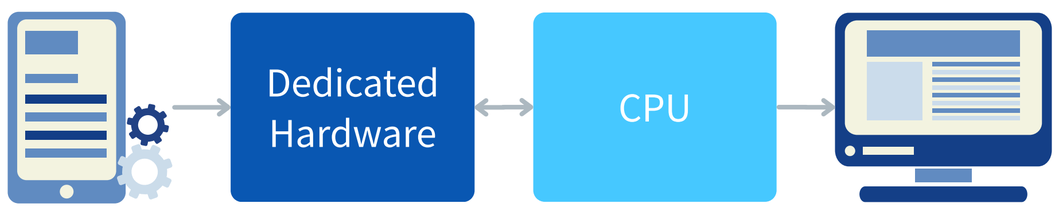
This project aims to explore hardware-based improvements in application-level emulation.

The proposed solution will:

- Have separate hardware handling the **translation**
- Have extensive support among **architectures**
- Remain **accessible** to most existing systems

We also aim to stimulate research into:

- Cross-architecture emulation
- Further hardware integration in emulators
- Multi-threaded programs



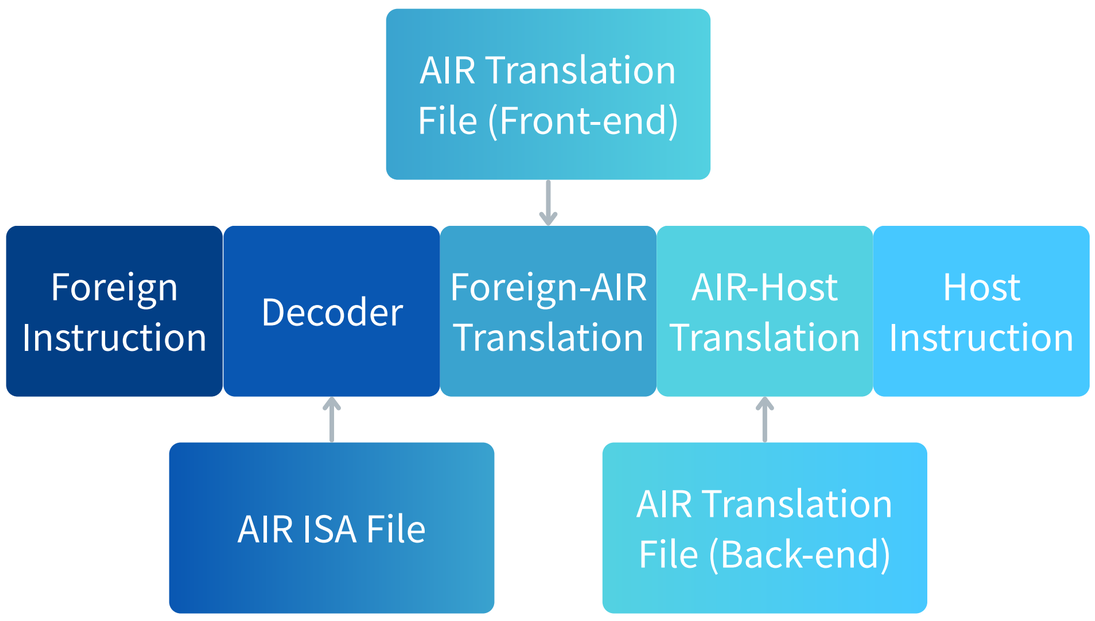
3 SOLUTION

Hardware acceleration is achieved through a PCIe **FPGA** chip, which can be configured at runtime to support a specific pair of architectures.

A new IR, Architecture-Independent Representation (**AIR**), is created to encode various Instruction Set Architectures (**ISA**s) and their instruction translation rules.

With AIR files, we can **generate** both a software simulator for testing purposes and a 3-stage **binary translator** to be implemented by the FPGA chip.

Finally, the FPGA chip will be integrated into a dynamic binary translator (DBT) similar to QEMU.

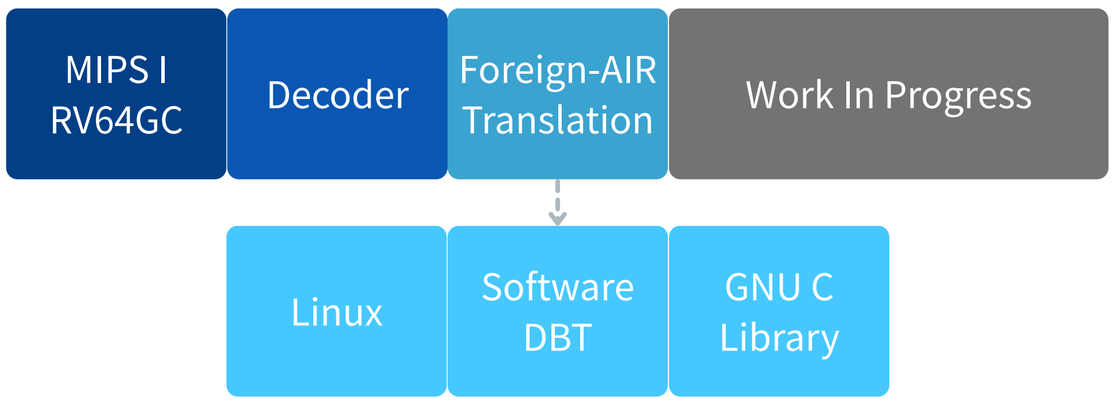


4 RESULT

We have created:

- AIR files for the RISC-V RV64GC and MIPS I ISA
- Generators for software decoders and translators
- Software-based DBT for testing purposes

The DBT can run statically linked C programs on Linux and perform arithmetic operations with standard I/O.



5 NEXT STEPS

We aim to:

- Implement the rest of RV64-x86 pipeline
- Generate FPGA designs
- Support code optimisation in hardware

CONTACT

sht2@st-andrews.ac.uk