Adding an accelerator to an AJIT processor system

Madhav P. Desai
Department of Electrical Engg.
IIT Bombay, Mumbai India

March 2, 2023

Overview

- Adding an accelerator to an AJIT system is easy.
- We assume that
  - The accelerator has a slave programming interface for configuration and monitoring.
  - The accelerator has a master memory interface to memory shared with the processor.
  - The accelerator generates an interrupt to the processor.

Our example accelerator

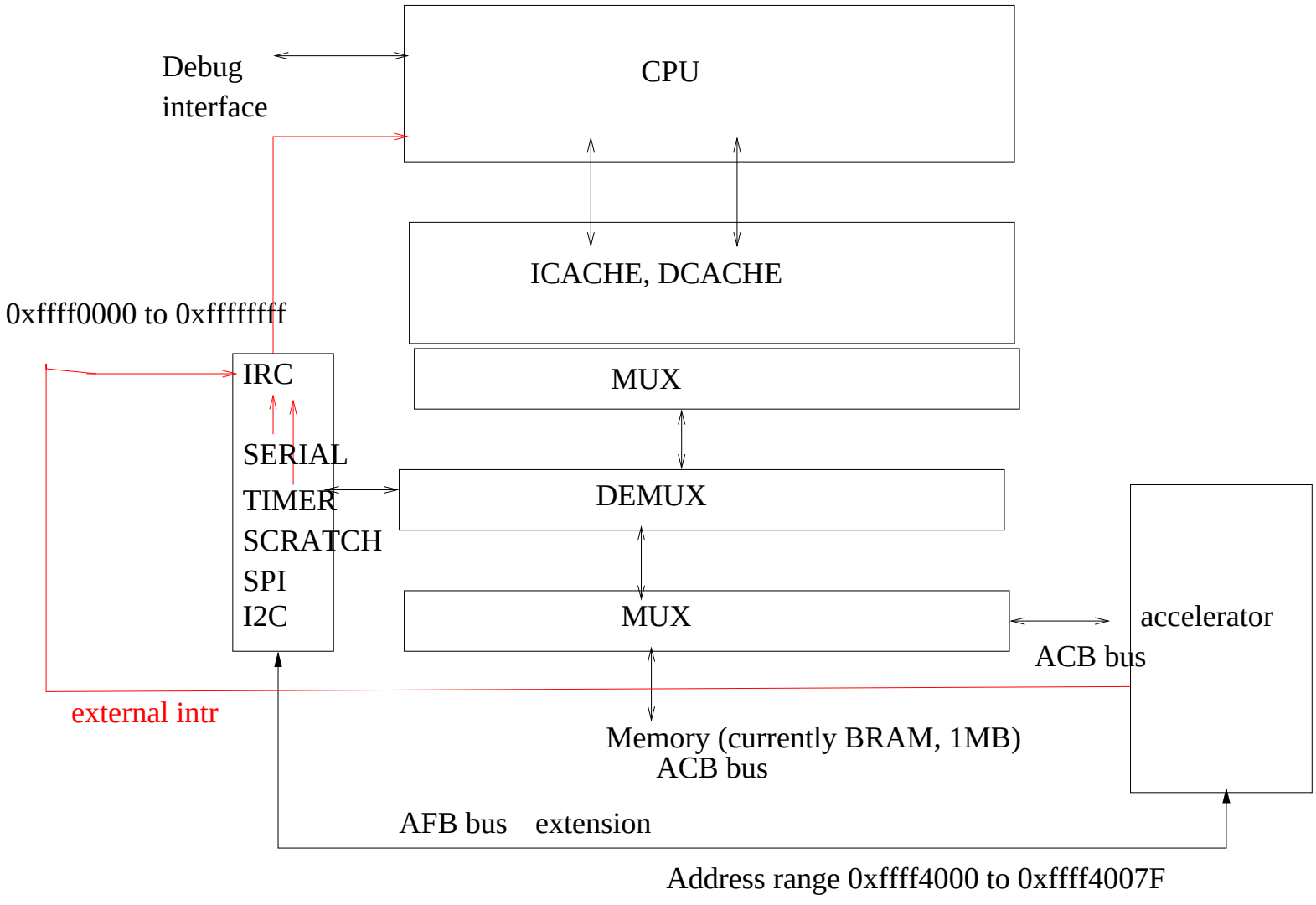
- The processor provides a 128 bit key to the accelerator.
- The processor indicates the region of memory to process.
- The processor indicates to the accelerator that it shall start processing.
- On starting, the accelerator computes an XOR of the key with 128-bit blocks in the region of memory specified by the processor.
- After finishing the job, the accelerator interrupts the processor.
- The processor should service the interrupt and move on.

Steps

- Fix the accelerator slave registers which can be controlled by the processor via the slave interface.
  - We use 16 32-bit registers, with register 0 mapped to address 0xffff4000. See README for register map.
- Implement the accelerator.
  - We do this using AHIR (Aa to VHDL).
  - Verify the accelerator using simulation.
- Integrate the accelerator with the processor to generate an FPGA top-level description.
- Synthesize.
- Write a test example to confirm the processor accelerator collaboration.

Integration of the accelerator with the processor

NOTES



Demo

- Take a look at the accelerator implementation in Aa.
- The VHDL simulation of the accelerator (generated by Aa to VHDL).
- The FPGA top-level.
- The synthesis script.
- The C test program for the processor, for checking the accelerator. Special attention to the interrupt handler.
- Run it all and check!

Summary

- End to end accelerator development flow.
- Accelerator interfaces: slave to processor, master to memory, interrupt.
- Easy to integrate: focus on accelerator implementation!