DTPM: Dynamic Trusted platform module

Dynamic Trusted Platform Module does run time integrity checking of the basic block instructions. This Defensive mechanism protects against TOCTOU attacks.

DTPM is implemented outside the processor pipeline to maintain simple and generic system that can be ported to any processor architectures. For this demonstration, we ported on to the openRISC architecture.

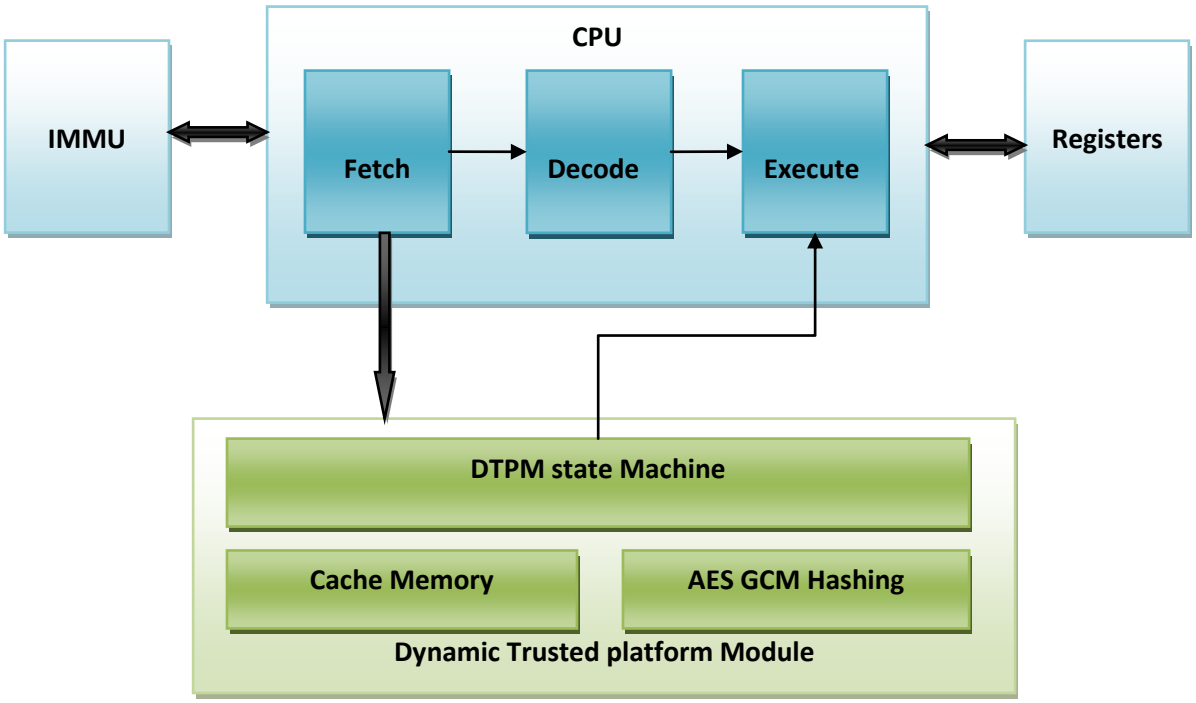
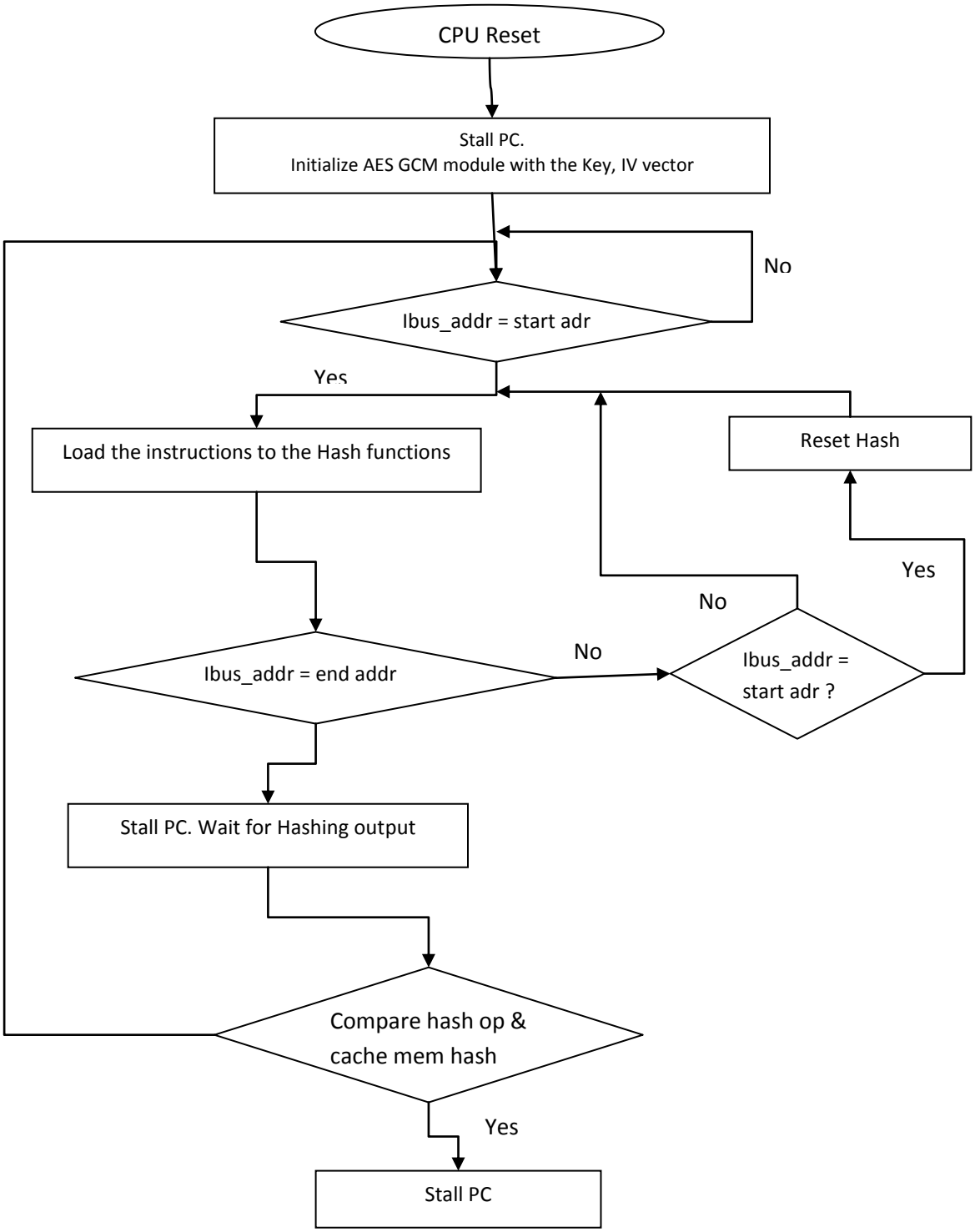


Figure 1: DTPM architecture

In openRISC architecture, CPU fetched instructions from IMMU through wishbone interface.

DTPM inputs are sampled on the wishbone interface which are driven by the Fetch module, and generates stall signal based on the basic block hash value. The input signals sampled are ibus_addr, ibus_addr_req, ibus_data, ibus_ack. The output signal is OR-ed with the external stall signal and driven to the execute stage.

State machine flow chart:



State Diagram:

1. Upon PC reset, DTPM stalls the PC and initializes AES GCM module.
2. DTPM monitors the address on the Fetch block and look for the Cache memory for start and end address of the basic blocks.
3. Once DTPM encounters the basic block start address, DTPM samples the instructions and loads into AES GCM. If two start addresses are encountered consecutively DTPM resets the AES GCM module considering it as a wrong branch prediction.
4. After DTPM encounters end address, DTPM loads last word into AES GCM module and stalls PC.
5. DTPM wait for the hash value from the AES GCM module.
6. Once Hash value is available, DTPM compares hash value to the cache memory basic block hash value.
7. If hash comparison passes DTPM resumes PC and proceed to the next basic block, else DTPM stalls PC.