SAFE Processor

Albert Kwon Udit Dhawan André DeHon

October 4, 2012

Figure 1 shows the overall micro-architecture of the SAFE processor. The figure shows all the functional units and the memory components in the current implementation of the SAFE processor.

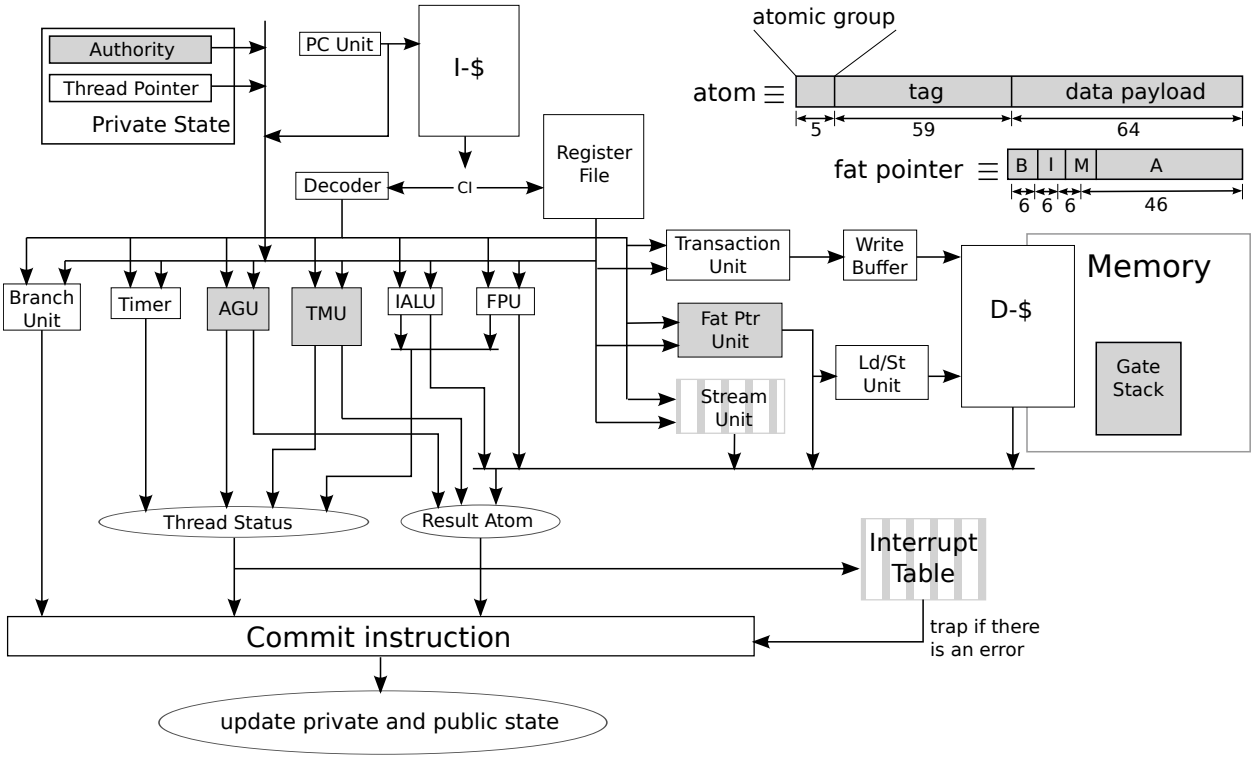

Figure 1: Functional Decomposition of Target SAFE Processor

1 Functional Units in SAFE

- **ALU:** regular 64b integer Arithmetic and Logic Unit
- **FPU:** regular double-precsion Floating Point Unit (not present in the current source code)
- **Branch Unit:** handles all the branching instructions
- **AGU:** Atomic Group Unit – validates the atomic groups (hardware data types) of the input operands for each instruction. See ISA Specification document for more details.
- **TMU:** Tag Management Unit – validates each instruction with respect to the tags on the various input fields as defined in the ISA Specification document and produces the output tags in case the instruction is validated. The various security policies are defined at the hardware level in form of rules composed using these tags.

  Top right portion of the Figure 1 shows the decomposition of a single word (called atom) in the SAFE processor. In the current implementation top 64 bits make up the atomic group and the tag for the word.
- **Fat Pointer Unit:** SAFE processor uses Fat Pointers for catching spatial violations in memory. This is performed by the Fat Pointer Unit.
- **Transaction Unit and Write Buffer:** SAFE ISA includes instructions for performing operations involving transactional behavior. These are performed using the Transaction Unit and the Write Buffer.

The current SAFE processor implementation is an unpipelined multi-phase design. In each processor cycle an instruction is fetched, decoded, and then executed using the appropriate functional units. In parallel with the operaiton, the instruction is validated for atomic group violation, spatial memory violations and TMU rules.