Short Description of Functional Simulator

Anuj Attri (aa11527)

The Python script is designed to simulate a vector core processor model. It encompasses several classes to manage instruction memory (**IMEM**), data memory for both scalar and vector data (**DMEM**), and a register file for scalar and vector registers. Each class is responsible for specific aspects of the simulation, including loading instructions and data from files, reading and writing to memory, and handling different data sizes and types.

- **IMEM**: Manages instruction memory. It loads instructions from a file and provides functionality to read instructions based on an index.
- **DMEM**: Manages data memory for both scalar (**SDMEM**) and vector (**VDMEM**) data. It supports reading and writing integer values from/to specified locations. The class ensures that operations stay within the bounds of 32-bit signed integer values.
- **RegisterFile**: Represents a register file containing a set number of registers, each of which can hold scalar or vector data. It supports reading and writing data to registers while ensuring data integrity.
- **Core**: Simulates the core of the vector processor. It interprets and executes instructions loaded into the IMEM, interacting with the DMEM and registers as needed. The core supports a variety of operations, including arithmetic, logical, memory access, control operations, and vector-specific instructions. It also handles vector mask and vector length registers for vector operations.

The script supports command-line arguments to specify the directory containing the input files (instructions and data). It initializes the memory and register file components with data from the specified files and then proceeds to execute the instructions.

The execution logic includes detailed handling for various instruction types:

- Vector and scalar arithmetic and logical operations
- Memory access operations, including load and store for scalar and vector data
- Control flow operations like branch instructions

- Special vector operations, such as manipulating the vector mask register and vector length register
- Register shuffle operations like PACKHI and UNPACKLO

To run the simulator, open and run (or double-click) the *aa11527_functional_simulator.bat* file. After executing the instructions, the script dumps the final state of the registers and memory to output files.