Stage 1: Design and testing of basic modules

The modules to be designed include ALU, Register File, Program Memory and Data Memory. For convenience, small sizes for memories are to be chosen for initial design stages.

ALU

It is a combinational circuit that takes two operands, does one of the 16 operations and produces a result. Operands and results are 32-bit std_logic_vectors. Further, there is a carry input and a carry output. The operation to be done is specified by another input that may be modelled as a 4-bit std_logic_vector, or more conveniently, as an enumerated type with DP opcodes as the set of symbols. Addition/subtraction requirements of ldr/str instructions can be handled without any additional circuitry by providing base and offset as operands and specifying the operation as add or sub. For incrementing PC and target address computation for branch instructions, separate modules will be used.

Register File 16x32

Register File contains an array of 16 std_logic_vectors of 32-bits each. Its inputs include two read addresses, one write address, one data input, one write enable and a clock. There are two data outputs on which contents of the array elements selected by read addresses are continuously available. If write enable is active, at clock edge the input data gets written in the array element selected by write address.

Program Memory and Data Memory 64x32 each

Each memory contains an array of 64 std_logic_vectors of 32-bits. Contents of Program Memory are initialized in declaration itself. Data Memory has one read port and one write port, wheras Program Memory has only a read port. Model read/write operations in same way as Register File, that is, write is clocked wheras read is unclocked (like a combinational circuit). Provide 4 write enable signals to provide byte level write operation.