The LM and SM

The LM\_SM block is responsible generating the control signals to run the Load and Store multiple instructions. Inherently, the instruction is a multi-cycle instruction and thus has to be performed by halting the pipeline.

The LM\_SM block has the following inputs and outputs:

Inputs

* 8-bit data (from the instruction to be fed to the priority encoder)
  + Note that there is a mux connected to it, since the inputs for LM and SM (which are the same) will be in different clock cycles, the details of which are explained later.
* LM bit and SM bit
  + The LM and SM bits specifically tell when the instruction is LM or SM and are used to activate the block, so that priority encoder can start giving address
* clk and reset

Outputs

* Register Address: AR2 in case of SM, AR3 in case of LM
* Clear and disable for the pipeline registers
* RF\_DO1\_mux: Controls the mux connected to the input of DO1 register in RR\_EX
* ALU2\_mux: Controls the mux connected to the second input of ALU, decides when +1 should be the input to the ALU
* AR3\_mux: Used in case of LM, this mux controls the data that is stored in AR3 in EX\_MM
* AR2\_mux: Used in case of SM, this mux controls the input to the register file for AR2
* mem\_in\_mux: Controls the input to the address of the memory which is usually the output of the ALU except in LM or SM where it is DO1

NOTE – The LM\_SM block starts outputting the address one cycle after it is activated

The block has been built using an FSM which goes into different states, depending on whether the instruction is LM or SM.

FSM Logic for LM

* The block gets activated when the current instruction is in the RF stage. Here it is in the S1 state. The bits that control memory input, AR3, and ALU are ‘1’ and are put through the pipeline register.
* It then moves to the S2 state, where the mux connected to input of DO1 now starts accepting the output of ALU (DO1 + 1), and the block starts outputting AR3 addresses, which will be written in the write back stage into the register file. The disable signal is high, since the registers are now disabled (RR\_EX, ID\_RR, IF\_ID and PC are disabled).
  + Note that there is a special enable signal to DO1 in RR\_EX since that has to be enabled during the LM\_SM process.
* The whole cycle continues till the last bit in the input of PE goes to zero, when the valid signal goes low, and one instruction currently in the RR\_EX register has to be disabled. The disable signal goes low as well, meaning the pipeline flow has started again.

Logic for SM –

* The block gets activated when the current instruction is in the decode stage.
* In the next clock cycle, the LM\_SM block starts outputting the AR2 address and thus, SM begins. The control bits follow the same pattern as LM, except the mux controlling the input to DO1 starts accepting the ALU output one cycle later. When the valid signal goes low, the last set of data is in the RR\_EX register and so, in SM, no clear signal is required. The disable signal goes low and pipeline resumes.