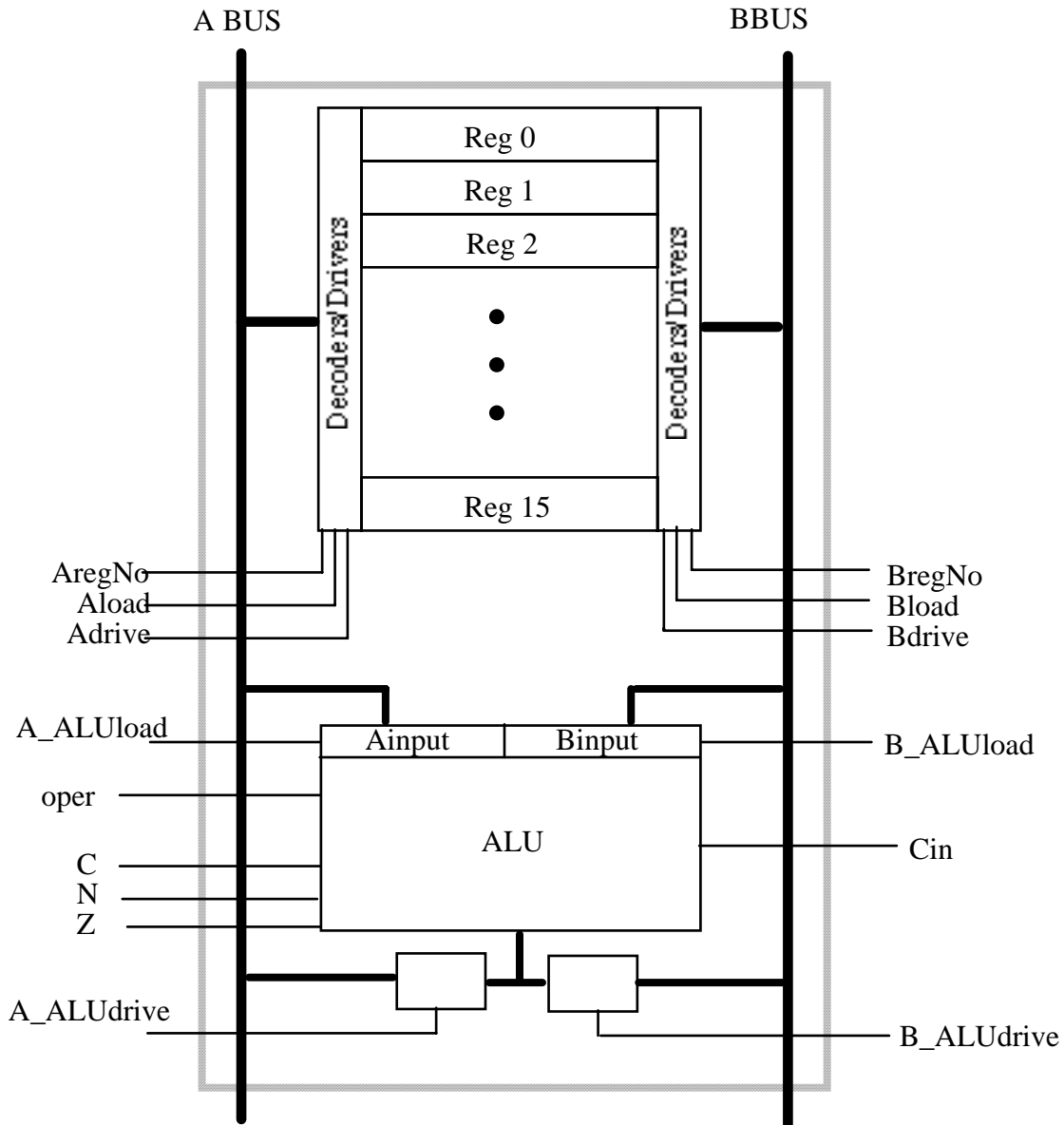


Project Assignment #8 DATAPATH DUE: Monday, May 12th

- 1) Copy the file **pr_step8.vhld** from ~degroat/ee762_assign. This file contains the testbench and the entity for the datapath description that you will write in this assignment.
- 2) The datapath that you will describe in VHDL is as follows:



The hashed perimeter shows the interface to the datapath. The interface contains

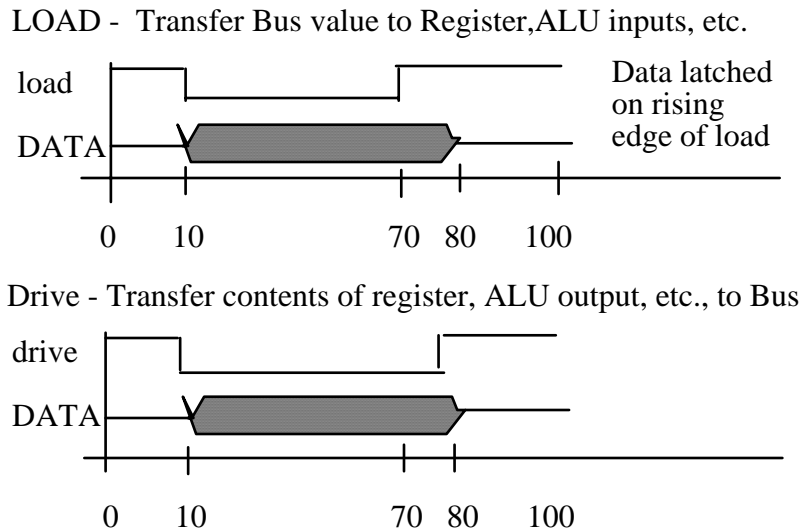
ABUS, BBUS - The two internal busses for moving information.

Aregload, Aregdrive, Aregno - The signals to control loading current Abus value into register Aregno, and driving contents of Aregno onto Abus

Bregload, Bregdrive, Bregno - The signals to control loading current Bbus value into register Bregno, and driving contents of Bregno onto Bbus

A_ALUload, B_ALUload - Control latching of Abus and Bbus value into the ALU inputs
 A_ALUdrive, B_ALUdrive - Drive results from ALU onto the respective bus.
 Cin - Carry into the ALU - Latched whenever either Ainp or Binp is latched.
 oper - Operation to be performed by ALU
 C,N,Z - flags for result of ALU operation - driven out when ALU results are driven onto one of the busses.

The timing of the datapath busses is the same as in step 7 for the registers.



Thus, to perform some operation with the ALU, on the first cycle the appropriate register number and register drive signal are asserted (set to '0'). At the same time the corresponding ALUload signal is asserted '0'. The result is that at 70 ns the ALU latches the value placed on the bus by the register. On the next cycle the ALU result is ready. The convention for the datapath is to always return the result on the Bbus. Thus the B_ALUdrive and Bload signals are asserted for transfer of the results back to the register.

Note that when initially loading values the testbench is driving the bus.

Also note that for two register operations both Abus and Bbus are driven by the registers during the first cycle.

3) As before, write a description of the datapath and simulate it. You will use the register set from step 7 and the ALU from step 6 (oper is of type operations). You will have to modify the bit vector length of the ALU so it is 16 bits.

Note that the buses are of type `std_logic_vector`, and Cin, C, N, and Z are of type `std_logic`. Use type conversion functions to convert from `std_logic_vector` to `bit_vector` and `std_logic` to `bit` for the inputs; `bit_vector` to `std_logic_vector` and `bit` to `std_logic` for the outputs.

4) Turn in

- a copy of the code (remember to turn in a copy of ALL CODE!!!) including packages if any.
- a copy of the simulation listing (use the `ps8_list.do` file) (remember – NO LINE WRAP!!!)
- a zoom->full_size waveform (again setup with the `ps8_wave.do` file)