EE375: COMPUTER ARCHITECTURE WITH MICROCONTROLLERS

PROJECT 1: Create a MARC2 Datapath in VHDL

50 Points, **Due Lesson 7** (at the start of class)

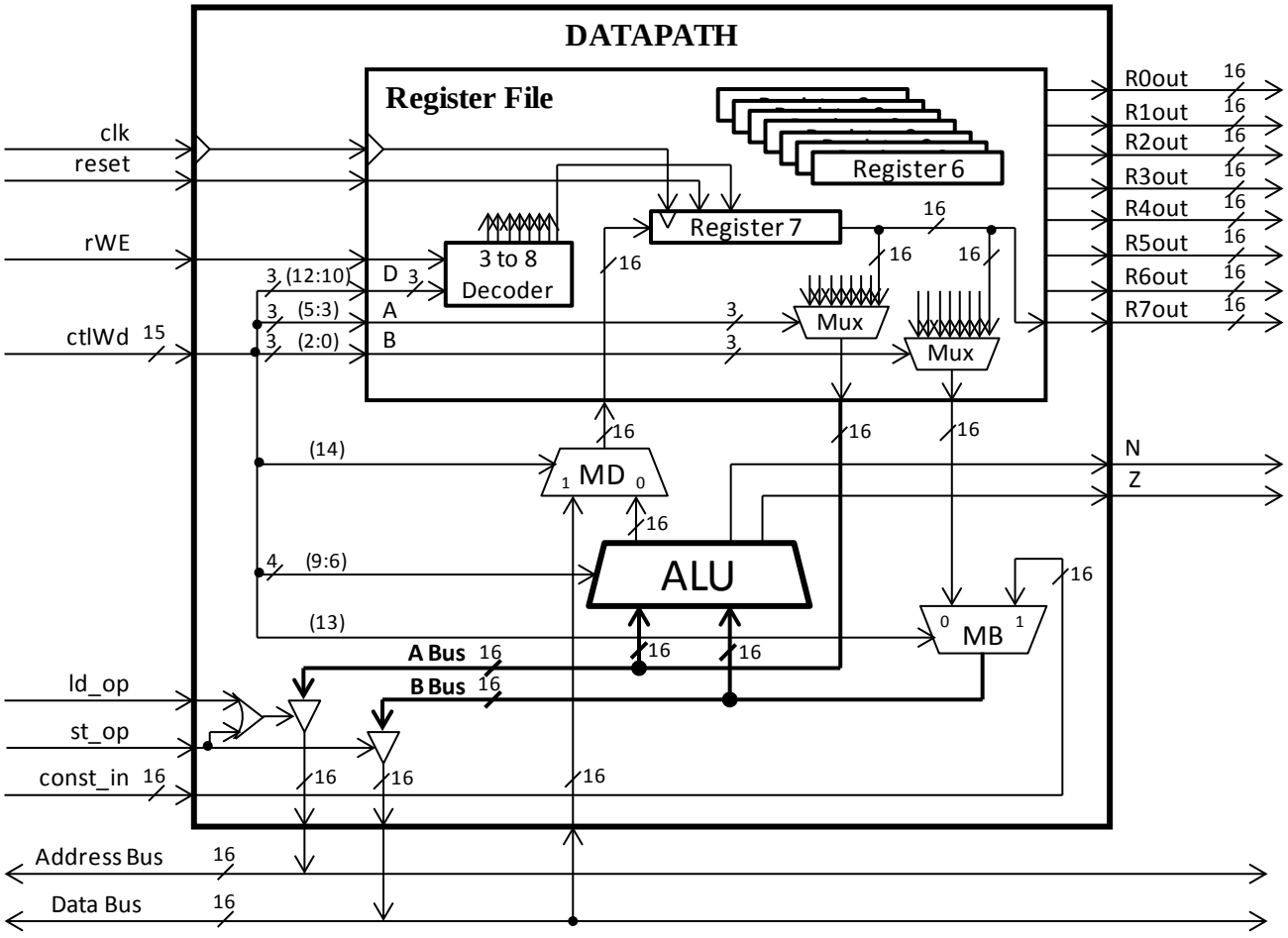
Learning Objective: In VHDL, design, test, and analyze a datapath.

Return (Paper copies must be submitted to your instructor and electronic copies must be saved to your Turn-in folder under the P1 subfolder NLT the start of your class period):

- *Electronic:* Copy of all VHDL files
- *Printed:* Test Plan (create your own)
- Annotated Simulation Screenshots (either electronic or hardcopy)
- *Printed:* Documentation as per course policy; annotate number of hours worked on cover sheet

Datapath Problem. Using Quartus, create a datapath component for a MARC2 processor using the shell file *datapath.vhd*. **Do not alter the entity declaration.** Add code/files to the architecture to provide the required functionality. After building your datapath, test it by simulating in ModelSim. You are encouraged to create your own incremental tests but you will also be evaluated on the correct execution of the provided testbench (*datapathTB.vht*). You will not need to implement this on an FPGA.

Overview: The MARC2 Datapath contains the hardware that takes an interpreted instruction and conducts the data processing tasks for that instruction. Instructions are fetched and decoded externally to the datapath (these are done in the control unit). Interaction with memory is done via system buses.



This block diagram contains the following four primary components:

A. **Register File:** This component contains 8 registers (labeled R0 thru R7). This component will output data from a selected register to the A and B Buses. It will receive data input from the ALU or the external Data Bus and store it in a selected register when enabled. Hardwire Register 0 to always store and output only 0's.

B. **Mulitiplexers:** The values asserted on the various internal and external busses must be selected through control signals (A, B, ctlWd(14), and ctlWd(13)).

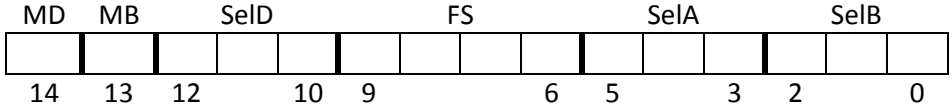
C. **Tri-State Buffers:** These control writes to the address and data buses based on the **ld_op**, **st_op**, and **ctlWd** signals. When these are at a state of high impedance (all 16 out bits at 'Z'), the datapath internal A and B buses are not connected to the system address and data buses.

D. **Function Unit (ALU):** Implement an ALU to accomplish the following functions:

S3	S2	S1	S0	RTL	Operation
0	0	0	0		None (nop); Assert high impedance
1	0	0	1	F ← A – B	A minus B (subtraction)
1	0	0	0	F ← A or B	A bitwise-OR B
0	1	1	1	$F \leftarrow B$	Pass B
0	1	1	0	$F \leftarrow A + B$	A plus B (addition)
0	1	0	1	$F \leftarrow A \wedge B$	A bitwise-AND B
0	1	0	0	$F \leftarrow \overline{A}$	Complement A
0	0	1	1	F ← srl A	Logical shift right of A
0	0	1	0	F ← sll A	Logical shift left of A
0	0	0	1	$F \leftarrow A$	Pass A

Operations on or loading immediate/constant values is possible via the **const_in** signal which is routed into the mux that selects the value to write to the B bus for input to the ALU or output to the system data bus. The datapath supplies two flags (**N** and **Z**) to indicate the status of the current ALU operation. The N-flag indicates if the sign (MSb) of the function unit output is negative (**N**=1 when the MSb is '1' from 2's complement arithmetic) and the Z-flag indicates if the result of a function unit operations is all 0's (**Z**=1).
NOTE: **use ieee.std_logic.unsigned.all;** do NOT use ieee.std_logic_arith.all to avoid typecasting issues.

The various control and selection inputs to the datapath are in the control word (the **ctlWd** input):



SelA selects which register places data on the A bus out of the register file. **SelB** similarly selects a register's contents from the register file but it must first be selected by **MB** against the **const_in** input to be placed on the B bus. **SelD** selects the destination register for a write action. **MD** selects whether the output of the ALU or a data word from memory goes to the destination register. **FS** determines the operation of the function unit (Function Unit Select) as listed in the truth table above. Other inputs to the datapath include the clock (**clk**) and the register file write enable (**rWE**). A register's value is updated when **rWE** = 1.