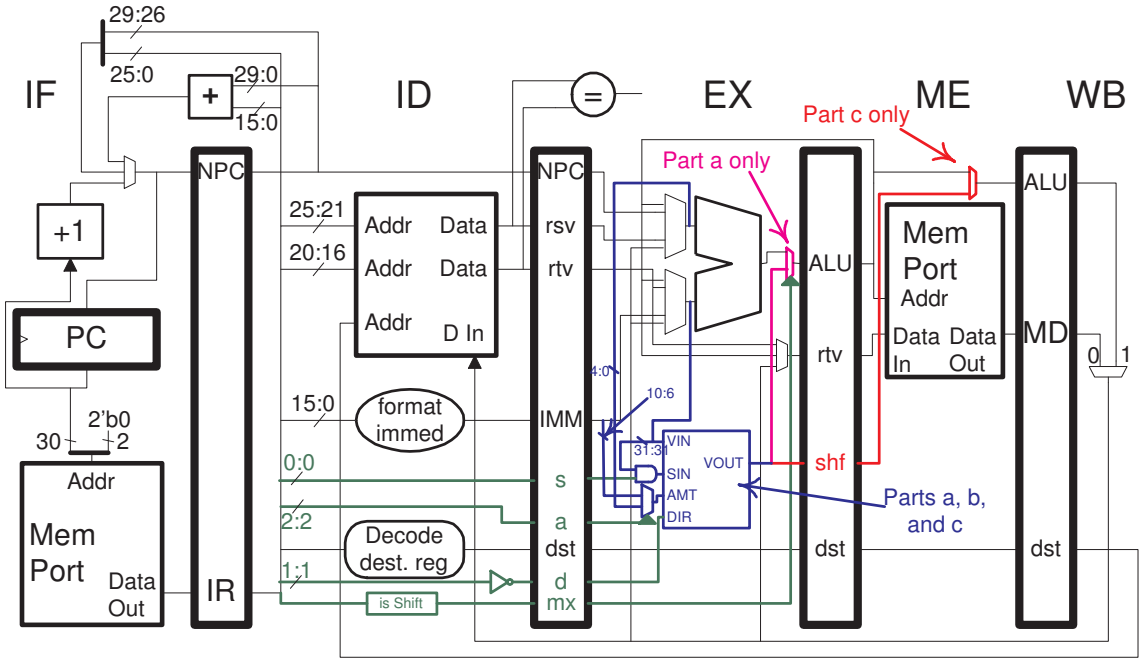
LSU EE 4720 **Homework 3** Solution **Due: 22 September 2010**

Problem 1: *Note: Problems like this one have been assigned before. Please solve this problem without looking for a solution elsewhere. If you get stuck ask for hints. Copying a solution will leave you unprepared for exams, and will waste your (or your parents') hard-earned tuition dollars.* A shift unit is to be added to the EX stage of the implementation below. The shift unit has a 32-bit data input, `VIN`, a 5-bit shift amount input, `AMT`, a 1-bit input `SIN`, and a 1-bit control input `DIR`. There is a 32-bit data output, `VOUT`. The `DIR` input determines whether the shift is left (1) or right (0). If the shift is right then the value at input `SIN` is shifted in to the vacated bit positions. The meaning of the other inputs is self-explanatory. For a description of MIPS-I instructions see the MIPS32 Volume 2 linked to the course references page.



(*a*) Connect the shift unit data inputs so that it can be used for the MIPS `sll`, `sllv`, `srl`, `srlv`, `sra`, and `srav` instructions. Assume that the ALU has plenty of slack (it is not close to carrying the critical path). (Control inputs are in the next part.)

- Be sure your design does not unnecessarily inflate cost or lower performance.
- In your diagrams be sure to use the bit ranges used, for example, 27:21, when connecting a wire to an input with fewer bits than the wire.

Solution appears above in blue and purple. The blue material applies to all parts.

The shift input and amount are taken from the ALU input mux outputs, this enables bypassing into the shift unit.

The shift output is muxed with the ALU output before the pipeline latch, so a new `EX/ME` latch is not needed.

Grading Note: There was no deduction for the following minor issue: Not using the bypassed values for shifter inputs (that would make stalls necessary in some cases).

(*b*) Show the logic for control inputs `DIR` and `SIN` and any multiplexors that you added.

Solution appears above in green. The control logic is simple due to the careful choice of function field values for the different shifts.

The box is Shift has an output of 1 if any of the shifts are present. If the implementation is of MIPS I then for the is Shift we can nor-together bits 31:26 (the opcode, to check for format R) and bits 5:3 (the function field, to test for a shift instruction). If MIPS IV is being implemented then this won't work because the output would be 1 for movt and movn instructions.

The shifts have been conveniently encoded so that if IR bit 0:0 is 1 then the shift is arithmetic, otherwise it is logical, that simplifies the SIN logic. Similarly bit 1:1 is 1 if the direction is right (but the problem was made up by a left-handed person, so an inverter is needed). A 1 in bit 2:2 indicates the shift amount comes from a register.

The s and a control signals are shown using their own pipeline latch bits, but it would be better to get their values from IMM, as is already done for the sa field value.

Grading Note: There was no deduction for the following minor issue: Computing the control signals in EX (doing that could lengthen critical path).

(*c*) Repeat the design of the datapath but assuming that the ALU is on the critical path and that we don't want to lower the clock frequency.

Solution appears in red. For part c, the ALU would once again be connected directly to the EX/ME.ALU latch. A new EX/ME.shf latch has been added to carry the shift value to ME where it enters a new mux. It is important that the output of this new mux NOT connect to the ME to EX bypass path (since the ALU is critical, as stated above) and that it not connect to the memory port Addr input (because the memory port is always assumed to be critical).