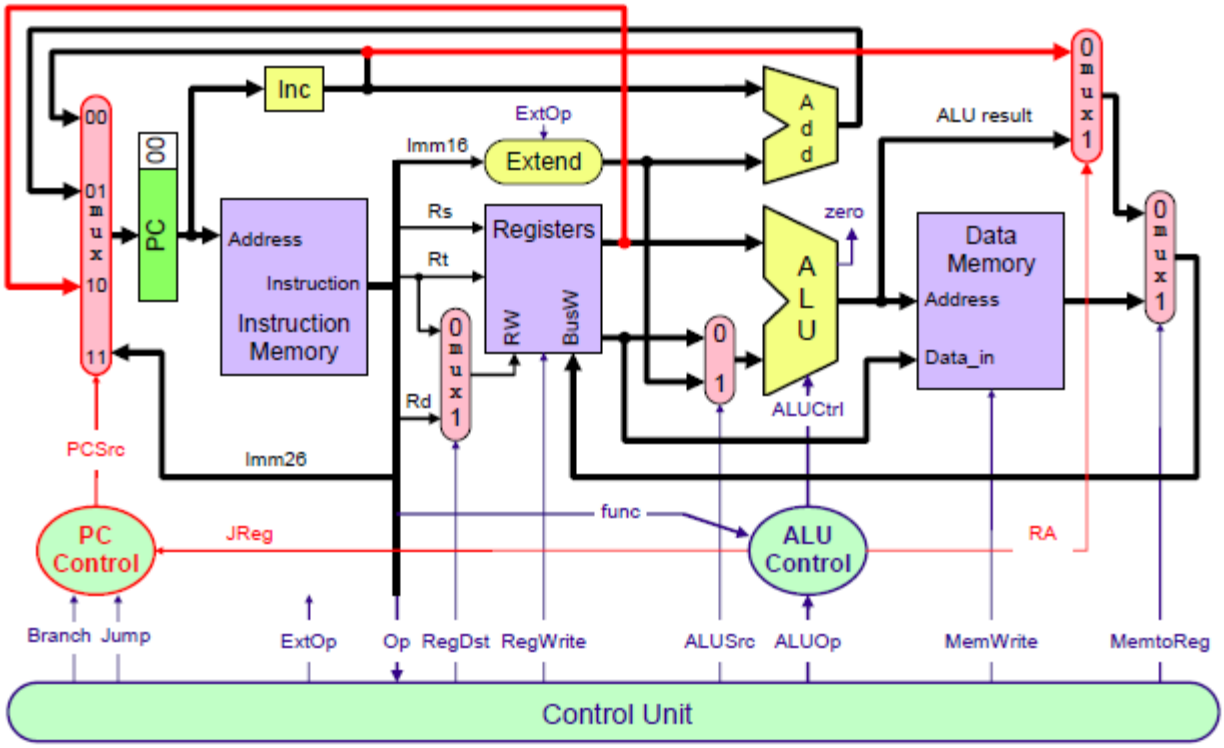
Q1. Single-Cycle MIPS Processor

We wish to add the instruction **jalr** (jump and link register) to the single-cycle datapath. The jump and link register instruction is described below:

```
jalr rd, rs # rd = pc + 4 , pc = rs
```

op[6] = 0	rs[5]	0	rd[5]	0	Funct[6] = 0x9

a) Add any necessary datapath and control signals and draw the result datapath. You should only add wires, gates, muxes to the datapath; do not modify the main functional units (the memory, register file, and ALU) themselves. Try to keep your diagram neat!



The necessary changes to the datapath and control:

For the datapath, we need a bigger 4-input multiplexer at the input of the PC. The first input is used to increment the PC. The second input is used for taken branches, where the branch target is PC-relative. The third input is used to jump register, where the input to the PC comes from a general-purpose register, and the fourth input is used for jump instructions.

For the implementation of the JALR instruction: to jump to register 'Rs', we need to add a path from the output of register Rs (first ALU input) back to the PC multiplexer input. PC control unit needs to be updated by adding an input control signal JReg (Jump Register) to select PC according to the value of register Rs. JReg is generated by the ALU control unit, since JALR is a R-

type instruction and JReg depends on the function field only. When JReg is equal to '1', PCSrc (PC control unit output control signal) will be '10' to select the value of register Rs as input to PC.

Also, we need to store PC+4 in register Rd. To accomplish this, we need another multiplexer to select between the incremented PC, the ALU result and data memory out, to be placed on BusW. Also, we need to add a path from the output of the incremented PC to the input of this new multiplexer. A control signal 'RA' (Return Address) is needed to select between the incremented PC and the ALU result. The MemtoReg multiplexer selects between the output of the 'RA' multiplexer and the Data Memory output to place on BusW.

b) Show the values of the control signals to control the execution of the **jalr** instruction. If you need add a new control signal, please add it along with its value to the table below. Use the following table for ALUCtrl.

ALU function	4-bit ALU Control
AND	0001
OR	0010
XOR	0011
ADD	0100
SUB	0101
SLT	0110

The main control signals for the JALR instruction are the same for other R-type instructions, such as ADD and SUB. The ALU Control signals for the JALR instruction require JReg = 1, RA = 0 and ALUCtrl is a don't care. These control signals are shown in the table below:

RegDst	RegWrite	ExtOp	ALUSrc	MemRead	MemWrite	MemtoReg	ALUCtrl	J	Beq	Bne	RA	JReg
Rd = 1	1	X	X	0	0	0	XXXX	0	0	0	0	1

Q2. Processor Performance

Suppose we add the multiply and divide instructions. The operation times are as follows:

Instruction memory access time = 190 ps, Data memory access time = 190 ps,
Register file read access time = 150 ps, Register file write access = 150 ps
ALU delay for basic instructions = 190 ps, ALU delay for multiply or divide = 550 ps
Ignore the other delays in the multiplexers, control unit, sign-extension, etc.