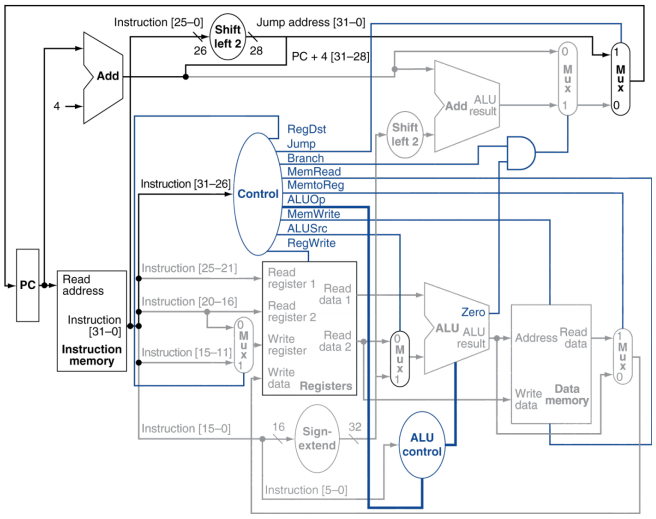
Computer Architecture 105_1

Homework #5 Single Cycle MIPS (PC+4 default version)

- Student ID: b03901032
- Name: 郭子生
- Goal: Implement a single cycle MIPS processor which supports several instructions listed below.



Instruction	Type	op code	func code
add	R	0	20
sub	R	0	22
and	R	0	24
or	R	0	25
slt	R	0	2A
lw	I	23	NA
sw	I	2B	NA
beq	I	4	NA
j	J	2	NA
jal	J	3	NA
jr	R	0	8

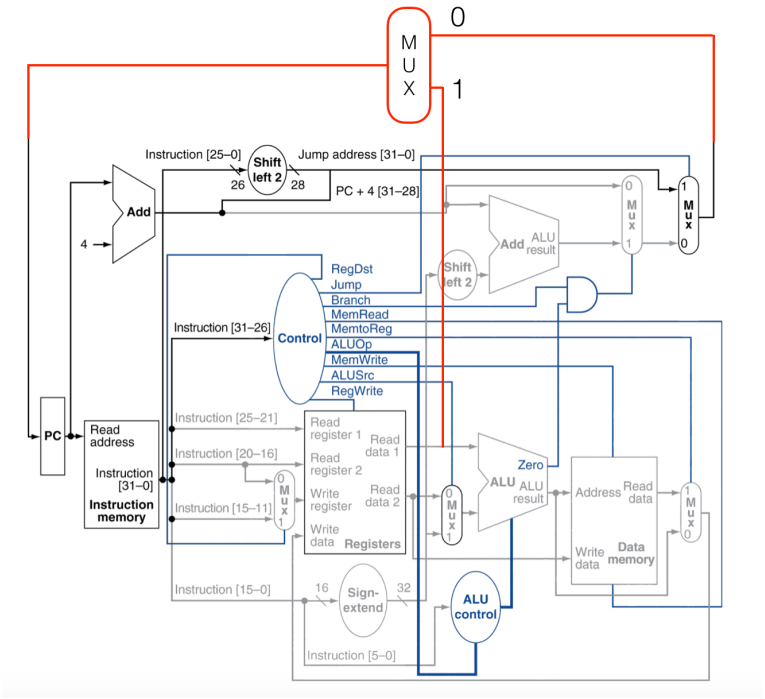
- Design:

Since there are numerous of components in the processor, my design is to construct each components in separate modules, then combine them together simply by connecting wires in the main processor module. The following are the modules which I had constructed:

Module Name	Description
PC	PC_o <= PC_i on the positive edge of CLK.
Adder	Output of an adder equals the sum of two inputs.
Registers	32 32-bits registers with several I/O pins. Write data into registers only on positive edge of CLK as well as the control signal RegWrite is true.
Control	Besides those I/O pins shown in the datapath above, I add one more input and two more outputs to support the instruction of jal and jr. The additional input is 6-bits funct from instruction and the additional outputs are control signal of JumpRegister and JumpAndLink.
ALU	Arithmetic logic unit supports operation such as ADD, SUB, AND, OR, and SLT. Also, if two input data are identical, the output ‘Zero’ will be true.
ALU Control	The module has two inputs of ALUOp and funct from instruction. The output is a 4-bits control signal to ALU.
Sign Extend Unit	Extend input data from 16 bits to 32 bits.
MUX	2-to-1 MUX with 1-bit select, two inputs can be any size.

• Special Hardware:

The datapath shown above doesn't support the instruction of jal and jr. Thus, additional hardware is required for there instructions. In my design, three additional MUXs are included. The first MUX is added as following so as to support instruction jr. The select signal of the MUX is JumpRegister signal from control unit. When JumpRegister is true, the next instruction address will be ReadData1; otherwise, it will remain as ordinary.



Another two MUXs are added before MUX_RegDST and after MUX_MemToReg separately. The first one has two inputs which are instruction [20:16] and integer 31, and with the select signal JumpAndLink from control unit. The second one has two inputs which are output of MUX_MemToReg and PC+4, and also with the select signal JumpAndLink form control unit. When JumpAndLink is true, the first MUX selects Reg 31 as Write register, and the second MUX selects PC+4 as Write data. Altogether, implement the function of instruction jal.