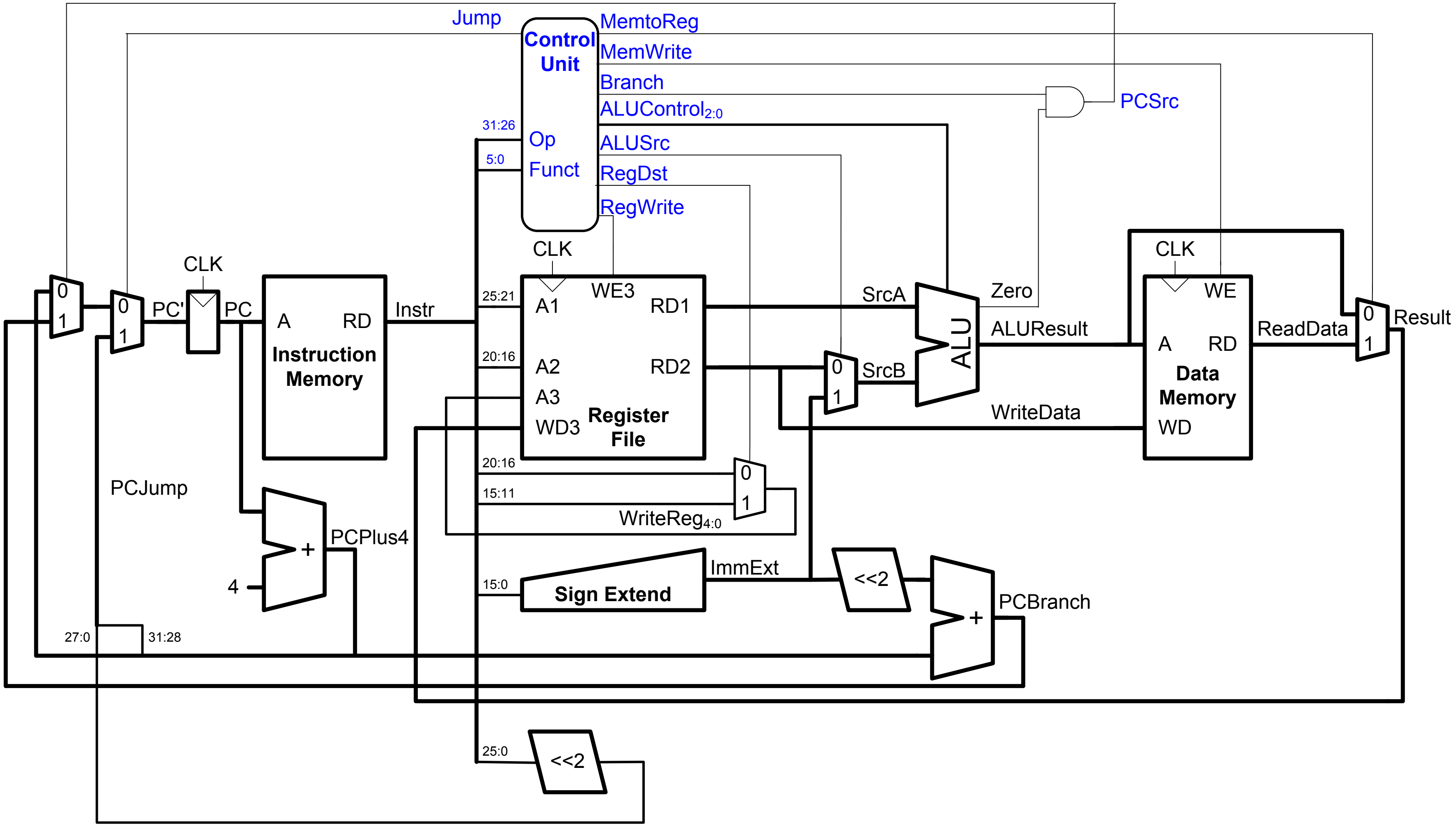
Cycle	reset	pc	instr	branch	srca	srcb	aluresult	zero	pcsrc	write data	mem write	read data
1	1	00	`addi $2,$0,5` `20020005`	0	0	5	5	0	0	x	0	0
2	0	04	`addi $7,$0,3` `20070003`	0	0	3	3	0	0			0
3	0	08	`addi $3,$0,0xc` `2003000c`	0	0	12	12	0	0			
4	0	0C										
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												

Table 1. First fourteen cycles of executing assembly program test1.asm

Remember, *branch* is asserted (1) when the instruction is a branch (`beq`) instruction. *aluout* is the output of the ALU at each cycle. *zero* is high (1) only if *aluout* is 0. *pcsrc*, a signal in the datapath, is low (0) when *nextpc* should be pc+4. *pcsrc* is high (1) when the *nextpc* should be the branch target address (*pcbranch*). You will notice that all of these signals are not available from the top-level module (`mips`). For debugging, you might want to look at these signals and others.



Single-cycle MIPS processor

Extended functionality. Main Decoder:

Instruction	**Op$_{5:0}$**	**RegWrite**	**RegDst**	**AluSrc$_{1:0}$**	**Branch**	**MemWrite**	**MemtoReg**	**ALUOp$_{1:0}$**	**Jump**			
R-type	000000	1	1	00	0	0	0	10	0			
`lw`	100011	1	0	01	0	0	1	00	0			
`sw`	101011	0	X	01	0	1	X	00	0			
`beq`	000100	0	X	00	1	0	X	01	0			
`addi`	001000	1	0	01	0	0	0	00	0			
`j`	000010	0	X	XX	X	0	X	XX	1			
`ori`	001101											
`bne`	000101											

Extended functionality. ALU Decoder:

ALUOp$_{1:0}$	**Meaning**
00	Add
01	Subtract
10	Look at `funct` field
11	