

Part II - Assemblers

In this part you will play the role of the assembler and assign a location counter (LC) value and a program counter (PC) for each instruction. In addition, you will actually assemble the given MIPS instructions by filling in the fields like the example provided in gray. This process will be reversed from what you implemented within Part I.

1. Provide the integer values for each of the fields in each MIPS instruction
2. Convert those integer values to their binary equivalent (32-bits)
3. Convert the 32-bit string to the hex value equivalent.

LC		addi	\$v0	\$zero	0		PC
		8	2	0	0		
001000 00000 00010 0000000000000000							
0	2002 0000						4
	next:	lw	\$t9	0	(\$a0)		
		35	25	0	4		
100011 00100 11001 0000000000000000							
4	8c990000						8
		addi	\$v0	\$v0	1		
		8	2	2	1		
001000 00010 00010 0000000000000001							
8	20420001						12
		sw	\$t9	\$0	(\$a1)		
		43	25	0	5		
101011 00101 11001 0000000000000000							
12	ACB9000						16
LC							PC
		addi	\$a0	\$a0	4		
		8	4	4	4		
001000 00100 00100 0000000000000100							

16	20840004						20
		addi	\$a1	\$a1	4		
		8	5	5	4		
001000 00101 00101 0000000000000100							
20	20A50004						24
		bne	\$t9	\$zero	next		
		5	25	0	-6		
000101 00000 11001 1111111111111010							
24	1419FFFA						28
		jr	\$ra				
		8	31				
000000 11111 0000000000000000 001000							
28	03E00008						32