SEAN CONNOR
24 June 2018

MIPS Assembly Workshop

Summer - 2018

This workshop exercise is to allow you to practice your skills at the art of assembling MIPS Assembly Language program code.

The following page has a small copy routine written in MIPS Assembly Language.

Please fill in the spaces in the table as my example shows:

Please use decimal values for the OpCode row, and Hex digits for the Instruction rows. Use your Green Card to lookup the required values.

Location
Counter

Program
Counter

		ADD	$t4	$s6	$s1	# $t4 = $s6 + $s1	
			12	22	17		
		Opcode	Rs	Rt	Rd	Immediate field	
		0 / 32	22	17	12		
	000000 10110 10001 01100 00000 100000						
32	02D1 6020						36

Additionally, use the Left margin to keep track of the Location Counter, and the right margin to note the value that would be in the Program Counter.

The initial value of the Location Counter should be zero.

The first instruction below is worked.

SEAN CONNOR

24 June 2018

LC		ADDI	$v0,	$zero,	0	# Initialize counter	PC
		8	2	0	0		
	001000 00000 00010 0000000000000000						
0	2002 0000						4
	next:	LW	$s1,	0	($a0)	# read the next word	
		35	17	0	4		
	100011 00100 10001 0000 0000 0000 0000						
4	8C910000						8
		ADDI	$v0,	$v0,	1	# count the copied word	
		8	2	2	1		
	001000 00010 00010 0000 0000 0000 0001						
8	20420001						12
		SW	$s1,	0	($a1)	# copy / store	
		43	17	0	5		
	101011 00101 10001 0000 0000 0000 0000						
12	ACB10000						16
		ADDI	$a0,	$a0,	4	# next source word	
		8	4	4	4		
	001000 00100 00100 0000 0000 0000 0100						
16	20840004						20
		ADDI	$a1,	$a1,	4	# next destination	
		8	5	5	4		
	001000 00101 00101 0000 0000 0000 0100						
20	20A50004						24
		BNE	$s1,	$zero,	next	# if not zero value	
		5	17	0	4	# See attached	
	000101 10001 00000 1111 1111 1111 1010						
24	1620FFFA						28
		JR	$ra			# return to caller	
		0	31				
	000000 11111 00000 0000 0000 0000 1000						
28	03E00008						32

SEAN CONNOR

24 June 2018

How to encode BNE jump to NEXT

current = LC + 4 = PC = 28
destination = LC = 4

$\Delta = 4 - 28 = -24$

$-24 \rightarrow$ 16 bit signed = 1111 1111 1110 1000

Need to align byte address so divide by 4 (shift right 2 bits)

$-24/4 = -6 =$ $\boxed{\text{1111 1111 1111 1010} = \text{0xFFFA}}$

This is the address relative to the BNE command. Looking at the instructions, this makes sense.

	1	ADDI	
NEXT	2	LW	6
	3	ADDI	5
	4	SW	4
	5	ADDI	3
	6	ADDI	2
	7	BNE	1
	8	JR	

We are moving six instructions "back", hence the negative value for the address.