Address(HEX)	标号	仿真平台2输入代码（这样输入才不会出错）	仿真平台2显示代码	机器码（BIN）	机器码（HEX）	指令类型	描述
0		lui x10, 0	lui a0, 0	00000000000000000001001101111	00000537	U	#initialize x10 =base address 0
4		ori x4, x10, 0	oritp, a0, 0	00000000000001010110001000010011	00056213	I	#x4<- base address x10 + offset 0 =0
8		addi x25, x0, 1	addi s9, x0, 1	00000000000100000000110010010011	00100c93	I	#initialize x25 = 1
c		addi x26, x0, 2	addi s10, x0, 2	00000000001000000000110110010011	00200d13	I	#initialize x26 = 2
10		addi x27, x0, 3	addi s11, x0, 3	00000000001100000000110110010011	00300d93	I	#initialize x27 = 3
14		addi x28, x0, 4	addi t3, x0, 4	00000000010000000000111000010011	00400e13	I	#initialize x28 = 4

				00010011			
18		sw x25, 0(x4)	sw s9, 0(tp)	00000001100100100010000010001111	01922023	S	#[0] = 1
1c		sw x26, 4(x4)	sw s10, 4(tp)	00000001101000100010001000111	01a22223	S	# [4] = 2
20		sw x27, 8(x4)	sw s11, 8(tp)	00000001101100100010010000111	01b22423	S	#[8] = 3
24		sw x28, 12(x4)	sw t3, 12(tp)	00000001110000100010011000111	01c22623	S	#[12] = 4
28		addi x5, x0, 4	addi t0, x0, 4	00000000010000000000010010011	00400293	I	# x5 = 4，循环次数
2c	Call：	Call: jal sum	jalra, 128	00000101010000000000000111011111	054000ef	UJ	# call function sum 跳转到 pc = 80

30		sw x12, 0(x4)	sw a2, 0(tp)	00000000110000100000100000001000	00c22023	S	#[16] <- 0x0000000a (x12=0x0000000a)
34		lw x19, 0(x4)	lw s3, 0(tp)	00000000000000100010100100000011	00022983	I	#x19<- [16]（0x10） ([16]=0x0000000a)
38		sub x18, x19, x12	sub s2, s3, a2	01000000110010011001001001100011	40c99833	R	#x18= 0
3c		addi x5, x0, 3	addi t0, x0, 3	00000000001100000000000010010011	00300293	I	#x5=3
40	loop2:	loop2:addi x5, x5, -1	addi t0, t0, -1	11111111111111001010000010010011	fff28293	I	# x5 -= 1
44		ori x18, x5, -1	ori s2, t0, -1	11111111111111001011101001000011	fff2e913	I	#x18= 0xffffffff ，（x18 = x5 or 12bit 立即数有符号扩展 0xffffffff）
48		xori x18, x18,	xori s2, s2,	010101	5559491	I	#X18=x18 ^ 1365

		1365	1365	01010110010100100100011	3		
4c		addi x19, x0, -1	addi s3, x0, -1	11111111111100000001001100100 11	fff00993	I	#X19=-1
50		andi x20, x19, -1	andi s4, s3, -1	11111111111110011111010000100 11	fff9fa13	I	#X20=0xffffffff ，（X20=0xffffffff and 0xffffffff）
54		or x16, x20, x19	or a6, s4, s3	00000001001110100110100000110011	013a6833	R	#X16=x20 | x19
58		xor x18, x20, x19	xor s2, s4, s3	00000001001110100100100100110011	013a4933	R	#X18=x20^ x19
5c		and x17, x20, x16	and a7, s4, a6	00000001000010100111000101100 11	010a78b3	R	#X17=x20 & x16
60		beq x5, x0, shift	beq t0, x0, 104	00000000000000101010	00028463	SB	#Ifx5 = 0 Goto shift after finished loop2 4

				00010001100011			times，goto pc= 68
64		j loop2	jal x0, 64	11111101110111111110000011011	fddff06f	UJ	#Loop Loop2 for 4 times，goto pc=40
68	shift:	shift:addi x5, x0, -1	addi t0, x0, -1	11111111111100000000010100100	fff00293	I	#X5=0xffffffff
6c		slli x18, x5, 15	slli s2, t0, 15	00000000111100101001100100011	00f29913	I	#X18=0xffff8000
70		slli x18, x18, 16	slli s2, s2, 16	00000001000010010001100100011	01091913	I	#X18= x18<< 16
74		srai x18, x18, 16	srai s2, s2, 16	01000001000010010101100100011	41095913	I	#X18=({32{a[31]}}<<(~b[16:0])|(a>>b[16:0])
78		srli x18, x18, 15	srli s2, s2, 15	00000000111110010101100100	00f95913	I	#X18=x18<<15

				11			
7c	finish:	finish:j finish	jal x0, 124	00000000000000000000000001101111	0000006f	UJ	#Endhere
80	sum:	sum:add x18, x0, x0	add s2, x0, x0	00000000000000000000000001101111	00000933	R	#X18 = x18+0
84	loop:	loop:lw x19, 0(x4)	lw s3, 0(tp)	00000000000000001000101001100000	00022983	I	#X19 <- [x4]
88		addi x4, x4,4	additp, tp,4	00000000010000100000100000100011	00420213	I	#x4 = x4 +4
8c		add x18, x18, x19	add s2, s2, s3	00000001001110010010010011001111	01390933	R	#X18= x18 + x19
90		addi x5, x5, -1	addi t0, t0, -1	11111111111111111001010000010100100111	fff28293	I	#x5 <- (x5-1)，循环次数-1

94		bne x5, x0, loop	bne t0, x0, 132	11111110000000101001100011100011	fe0298e3	SB	#loop 循环累加次，结果存于：x5
98		slli x12, x18, 0	slli a2, s2, 0	00000000000010010001011000100011	00091613	I	#X12<- x18 ，X12 = 0x0000000a，函数调用结果存于：x12
9c		Jr ra	jalr x0, 0(ra)	00000000000000001000000011001111	00008067	I	#函数 sum 调用返回，回到 pc = 84