1.

|  |  |  |  |
| --- | --- | --- | --- |
| MIPS | Syntax | binary | hex |
| Loop: sw $t1, 4($s0) | SW rt, offset(base)  SW = 101011  rt = 01001  base = 10000  offset(4) = 0000000000000100 | 101011 01001 10000 0000000000000100 | 0xAE090004 |
| addi $t1, $t1, -1 | ADDI rt, rs, immediate  ADDI = 001000 rt = 01001 rs = 01001 immediate = 1111111111111111 | 001000 01001 01001 1111111111111111 | 0x2129FFFF |
| sll $t1, $t1, 2 | SLL rd, rt, sa  SPECIAL - 000000 SLL - 000000 0 - 00000 rd - 01001 rt - 01001 sa - 00010 | 000000 00000 01001 01001 00010 000000 | 0x00094880 |
| bne $t1, $s5, Exit | BNE rs, rt, offset  BNE - 000101 rs - 01001 rt - 10101 offset - 0000000000011000 | 000101 01001 10101 0000000000011000 | 0x15350018 |
| addi $s0, $s0, 4 | ADDI rt, rs, immediate  ADDI - 001000 rt - 10000 rs - 10000 immediate - 0000000000000100 | 001000 10000 10000 0000000000000100 | 0x22100004 |
| j Loop | J target  J - 000010 target - 11111111110000000000000000 | 000010   11111111110000000000000000 | 0x0BFF0000 |

2. ADD $s0, $s0, $s0

3. ffffffab = 1111 1111 1111 1111 1111 1111 1010 1011

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1111** | **1111** | **1111** | **1111** | **1111** | **1111** | **1010** | **1010** |
|  | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0101 | 0100 |
| + |  |  |  |  |  |  |  | 1 |
|  | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0101 | 0101 |

(20 \* 1) + (21 \* 0) + (22 \* 1) + (23 \* 0) + (24 \* 1) + (25 \* 0) + (26 \* 1) + (27 \* 0) = 85

4.

Loop: if (i <100)

{ i = 4 \* i;

A[i] = 0;

goto Loop;}

=>

i = 0;

while (i < 100) do

{ i = 4 \*i; A[i] = 0}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | and | $s0, | $s0, | $zero | # let i = 0 |
|  | andi | $t1, | $zero, | 100 | # intermediate 100 |
| Loop: | lw | $t0, | 0($s1) |  | # temp1 = A[i] |
|  | bge | $s0, | $s2, | exit | # branch if ! (i < 100) |
|  | sll | $s0, | $s0, | 2 | # mult by 4 |
|  | bne | $s0, | $s2, | Loop | # if i < 100 |
| Exit: |  |  |  |  |  |