Mark Gameng

Lab 1

Step 2:

|  |  |
| --- | --- |
| Input | Output |
| 1 | 1 |
| 2 | 2 |
| 3 | 6 |
| 4 | 24 |
| 5 | 120 |
| 6 | 720 |

Step 3: The output is the factorial of the input, 1! = 1, 2! = 2, 3! = 6, 4! = 24, 5! = 120, 6! = 720

Step 4:

|  |  |
| --- | --- |
| Label | Address |
| main | 0x00400000 |
| loop | 0x00400028 |
| result | 0x00400034 |
| end | 0x00400054 |
| prompt | 0x10010000 |
| prompt2 | 0x10010040 |
| bye | 0x10010050 |

Step 5:

|  |  |  |  |
| --- | --- | --- | --- |
| Address | Machine Instructions (code) | Assembly statements (basic) | Source statements (pseudo instructions) |
| 0x00400018 | 0x1840000e | blez, $2, 0x0000000e | 12: blez $v0, end  # branch to end if $v0 < = 0 |
| 0x0040001c | 0x24080000 | addiu $8,$0,0x00000000 | 13: li $t0, 0  # clear register $t0 to zero |
| 0x00400020 | 0x00024021 | addu $8,$0,$2 | 15: move $t0,$v0  # assign $v0 to $t0 |
| 0x00400024 | 0x24090001 | addiu $9,$0,0x00000001 | 16: li $t1,1 |
| 0x00400028 | 0x71284802 | mul $9,$9,$8 | 18: mul $t1 $t1 $t0  # $t1 = $t1 \* $t0 |
| 0x0040002c | 0x2108ffff | addi $8,$0,0xffffffff | 19: addi $t0 $t0 -1  # summing integers with -1 |
| 0x00400030 | 0x1500fffd | bne $8,$0,0xfffffffd | 20: bnez $t0 loop  # branch to loop if $t0 is != zero |
| 0x00400034 | 0x24020004 | addiu $2,$0,0x00000004 | 23: li $v0,4  # system call code for Print String |
| 0x00400038 | 0x3c011001 | lui $1,0x00001001 | 24: la $a0, prompt2  # load address of prompt2 into $a0 |
| 0x0040003c | 0x34240040 | ori $4,$1,0x00000040 |  |

Step 6:

The size of an instruction in bytes is 4

The size of an address in bytes is 4

Step 7:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address (PC) | R4 [a0] | R2[v0] | R8[t0] | R9[t1] | Source statements |
| 0x00400000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | li $v0,4 |
| 0x00400004 | 0x00000000 | 0x00000004 | 0x00000000 | 0x00000000 | la $a0, prompt |
| 0x00400008 | 0x00000000 | 0x00000004 | 0x00000000 | 0x00000000 |  |
| 0x0040000c | 0x10010000 | 0x00000004 | 0x00000000 | 0x00000000 | syscall |
| 0x00400010 | 0x10010000 | 0x00000004 | 0x00000000 | 0x00000000 | li $v0,5 |
| 0x00400014 | 0x10010000 | 0x00000005 | 0x00000000 | 0x00000000 | syscall |
| 0x00400018 | 0x10010000 | 0x00000003 | 0x00000000 | 0x00000000 | blez $v0, end |
| 0x0040001c | 0x10010000 | 0x00000003 | 0x00000000 | 0x00000000 | li $t0, 0 |
| 0x00400020 | 0x10010000 | 0x00000003 | 0x00000000 | 0x00000000 | move $t0,$v0 |
| 0x00400024 | 0x10010000 | 0x00000003 | 0x00000003 | 0x00000000 | li $t1,1 |
| 0x00400028 | 0x10010000 | 0x00000003 | 0x00000003 | 0x00000001 | mul $t1 $t1 $t0 |
| 0x0040002c | 0x10010000 | 0x00000003 | 0x00000003 | 0x00000003 | addi $t0 $t0 -1 |
| 0x00400030 | 0x10010000 | 0x00000003 | 0x00000002 | 0x00000003 | bnez $t0 loop |
| 0x00400028 | 0x10010000 | 0x00000003 | 0x00000002 | 0x00000003 | mul $t1 $t1 $t0 |
| 0x0040002c | 0x10010000 | 0x00000003 | 0x00000002 | 0x00000006 | addi $t0 $t0 -1 |
| 0x00400030 | 0x10010000 | 0x00000003 | 0x00000001 | 0x00000006 | bnez $t0 loop |
| 0x00400028 | 0x10010000 | 0x00000003 | 0x00000001 | 0x00000006 | mul $t1 $t1 $t0 |
| 0x0040002c | 0x10010000 | 0x00000003 | 0x00000001 | 0x00000006 | addi $t0 $t0 -1 |
| 0x00400030 | 0x10010000 | 0x00000003 | 0x00000000 | 0x00000006 | bnez $t0 loop |
| 0x00400034 | 0x10010000 | 0x00000003 | 0x00000000 | 0x00000006 | li $v0,4 |
| 0x00400038 | 0x10010000 | 0x00000004 | 0x00000000 | 0x00000006 | la $a0, prompt2 |