CS24 Elementary Computer Organization
Chapter 2 Exercises: 2.3, 2.7, 2.13, 2.18

2.3 [5] <2.2, 2.3> For the following C statement, write the corresponding RISC-V assembly code. Assume that the variables f, g, h, i, and j are assigned to registers x5, x6, x7, x28, and x29 respectively. Assume that the base address of the arrays A and B are in registers x10 and x11, respectively.

Variable	Numeric Register	Human Equivalent
f	x5	t0
g	x6	t1
h	x7	t2
i	x28	t3
j	x29	t4
A	x10	a0
B	x11	a1

C Code:

B[8] = A[i - j];

Equivalent in RISC-V:

```
sub t0, t3, t4      # f = i - j
slli t0, t0, 2      # f = (i - j) * 4
add t1, a0, t0      # g = & A[f]
lw t2, 0(t1)        # h = A[f]
sw t2, 32(a1)       # B[8] = h
```

2.7 [5] <2.2, 2.3> Translate the following C code to RISC-V. Assume that the variables f, g, h, i, and j are assigned to registers x5, x6, x7, x28, and x29 respectively. Assume that the base address of the arrays A and B are in registers x10 and x11, respectively. Assume that the arrays of A and B are 8-byte words.

Variable	Numeric Register	Human Equivalent
f	x5	t0
g	x6	t1
h	x7	t2
i	x28	t3
j	x29	t4
A	x10	a0
B	x11	a1

C Code:

```
B[8] = A[i] + A[j];
```

Equivalent in RISC-V:

```
slli t3, t3, 2      # i = i * 4
slli t4, t4, 2      # j = j * 4
add t0, a0, t3      # f = & A[i]]
add t1, a0, t4      # g = & A[j]]
lw t0, 0(t0)        # f = A[i]
lw t1, 0(t1)        # g = A[j]
add t0, t0, t1      # f = A[i] + A[j]
sw t0, 32(a1)       # B[8] = f
```

2.13 [5] <2.2, 2.5>Provide the instruction type and hexadecimal representation of the following instruction:

Instruction:

sw x5, 32(x30)

This is a Type-S Instruction:

6 bits	5 bits	5 bits	3 bits	5 bits	7 bits
immediate [11 : 5]	rs2	rs1	funct3	immediate[4:0]	opcode

Fill in with instruction quantities:

6 bits	5 bits	5 bits	3 bits	5 bits	7 bits
32 [11 : 5]	x5	x30	010	32 [4:0]	0100011

Binary representation of quantities:

6 bits	5 bits	5 bits	3 bits	5 bits	7 bits
0000001	00101	11110	010	00000	0100011

Break binary line into sections of 4 bits for hexadecimal translation:

0000	0010	0101	1111	0010	0000	0010	0011
0	2	5	F	2	0	2	3

Representation of the above instruction in hexadecimal:

025F2023

2.18 [10] <2.6>Find the shortest sequence of RISC-V instructions that extracts bits 16 down to 11 from register x5 and uses the value of this field to replace bits 31 down to 26 in register x6 without changing the other bits of registers x5 or x6. (Be sure to test your code using x5 = 0 and x6 = 0xffffffffffffffff. Doing so may reveal a common oversight.)

First Register	x5
Second Register	x6
Bit Mask Register	t0

```
lui t0, 0x0000F      # Load upper 20 bits of bitmask. t0 = 0x0000F000
addi t0, t0, 0xC00 # Add lower 12 bits of bitmask. t0 = 0x0000FC00
and t0, x5, t0       # Copy x5[11 - 16] into t0
slli t0, t0, 16      # Move t0[11 - 16] to t0[27 - 32]
slli x6, x6, 6       # Remove x6[27-32] to free up space
srli x6, x6, 6       # Shift bits back into original position
or x6, x6, t0        # Copy t0[27-32] into x6[27-32]
```