

if-else Statement

Convert the following C++ code to LEGv8 Assembly code. Assume the variables f, g, h, i, and j correspond to five registers X19, X20, X21, X22, and X23.

```
if (i == j)
    f = g + h;
else
    → f = g - h;
```

X19 X20 X21 X22 X23
f g h i j

CMP X22, X23

// compare i and j

B.NE L1

// if i ≠ j, Go to L1.

→ ADD X19, X20, X21

// f = g + h

L1 : SUB X19, X20, X21

// X19 = X20 - X21
f = g - h

Convert the following C++ code to LEGv8 Assembly code. Assume the variables i and k correspond to registers X22 and X24. The base address of the array save is in X25.

```
while (save[i]==k)
{
    i=i+1;
}
```

X22 X24 X25
i k save

Loop:

LSL X10, X22, #3

// $X10 = X22 \times 2^3$
 $= i \times 8$

ADD X10, X10, X25

// $X10 = X25 + i \times 8$

LDUR X9, [X10, #0]

// Load save[i]
// $X9 = \text{save}[i]$

CMP X9, X24

// compare save[i] and k

B.NE Exit

// if $\text{save}[i] \neq k$, Go to Exit

ADDI X22, X22, #1, // $i = i + 1$

B Loop

Exit:

While loop

Convert the following C++ code to LEGv8 Assembly code. Assume the variables i and k correspond to registers X22 and X24.

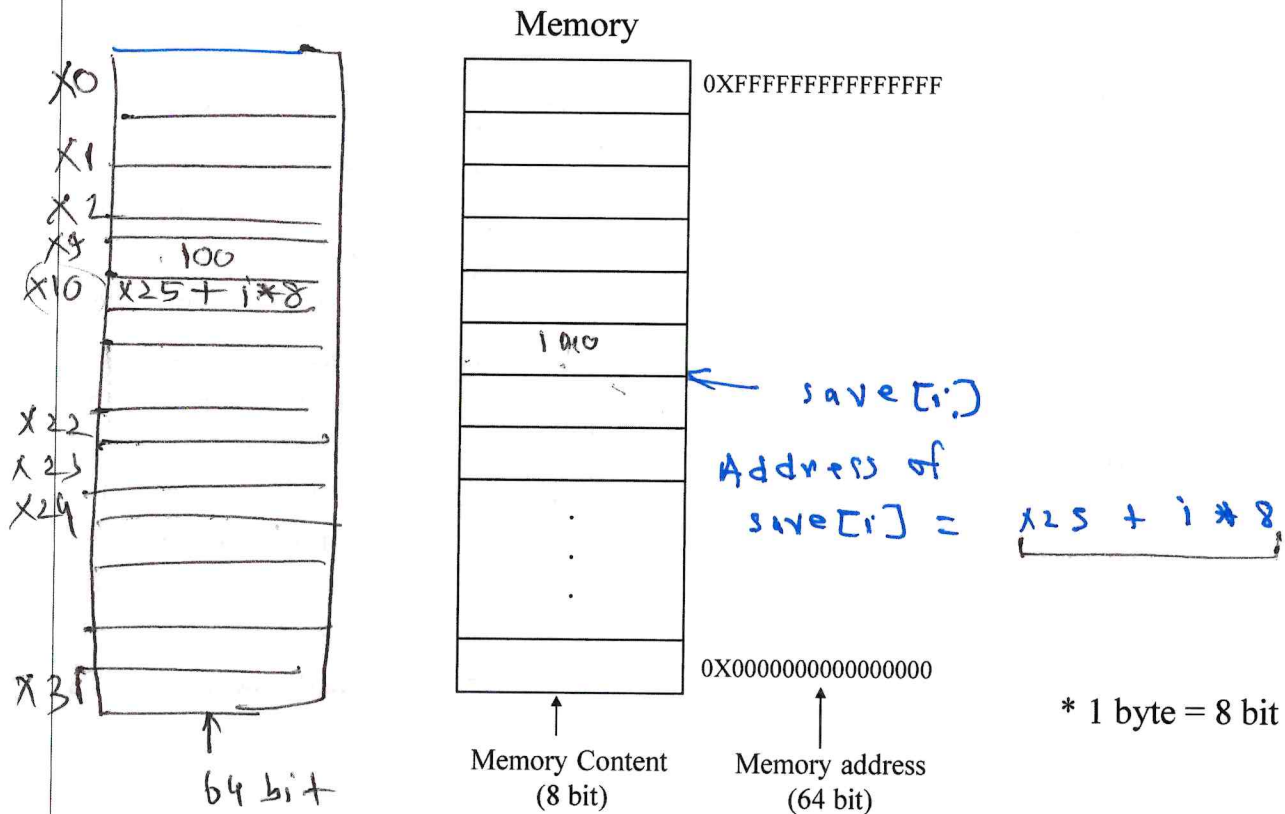
```
while (i==k)
{
    i=i+1;
}
```

X22, X24
↓ ↓
i k

```
Loop:  CMP    X22, X24           // compare i and k
      B.NE    Exit              // if i ≠ k, Go to exit
      ADDI    X22, X22, #1      // i = i + 1
      B       Loop              // Go to Loop.

Exit:
```

Memory



Problem: Assume that A is an array of 4 integer type elements (10, 20, 15, 8). Each element is 64 bit (doubleword). How does the array elements contained on the memory? The base address of the array A is 0X0000000000000000.

SOLUTION:

int A = {10, 20, 15, 8};

A				
10	20	15	8	→ Elements
0	1	2	3	→ index

A[0] = 10 = 0X000000000000000A

A[1] = 20 = 0X0000000000000014

A[2] = 15 = 0X000000000000000F

A[3] = 8 = 0X0000000000000008

ADDI X9, XZR, #0 // i = 0

for Loop

Convert the following C++ code to LEGv8 Assembly code. Assume the variable a is in X22 and base address of array b is in X23.

```
for(i=0, i<a, i++)
{
    b[i] = a + i;
}
```

X22 X23
↓ ↓
a b[]
assume X9
↓

MOVI X9, #0

// i = 0

Loop: CMP X9, X22

// compare i and a

B.GE Exit

ADD X10, X22, X9

// X10 = a + i

LSL X11, X9, #3

// X11 = X9 * 8 = i * 8

ADD X11, X23, X11

// X11 = X23 + i * 8

STUR X10, [X11, #0]

// b[i] = a + i

ADDI X9, X9, #1

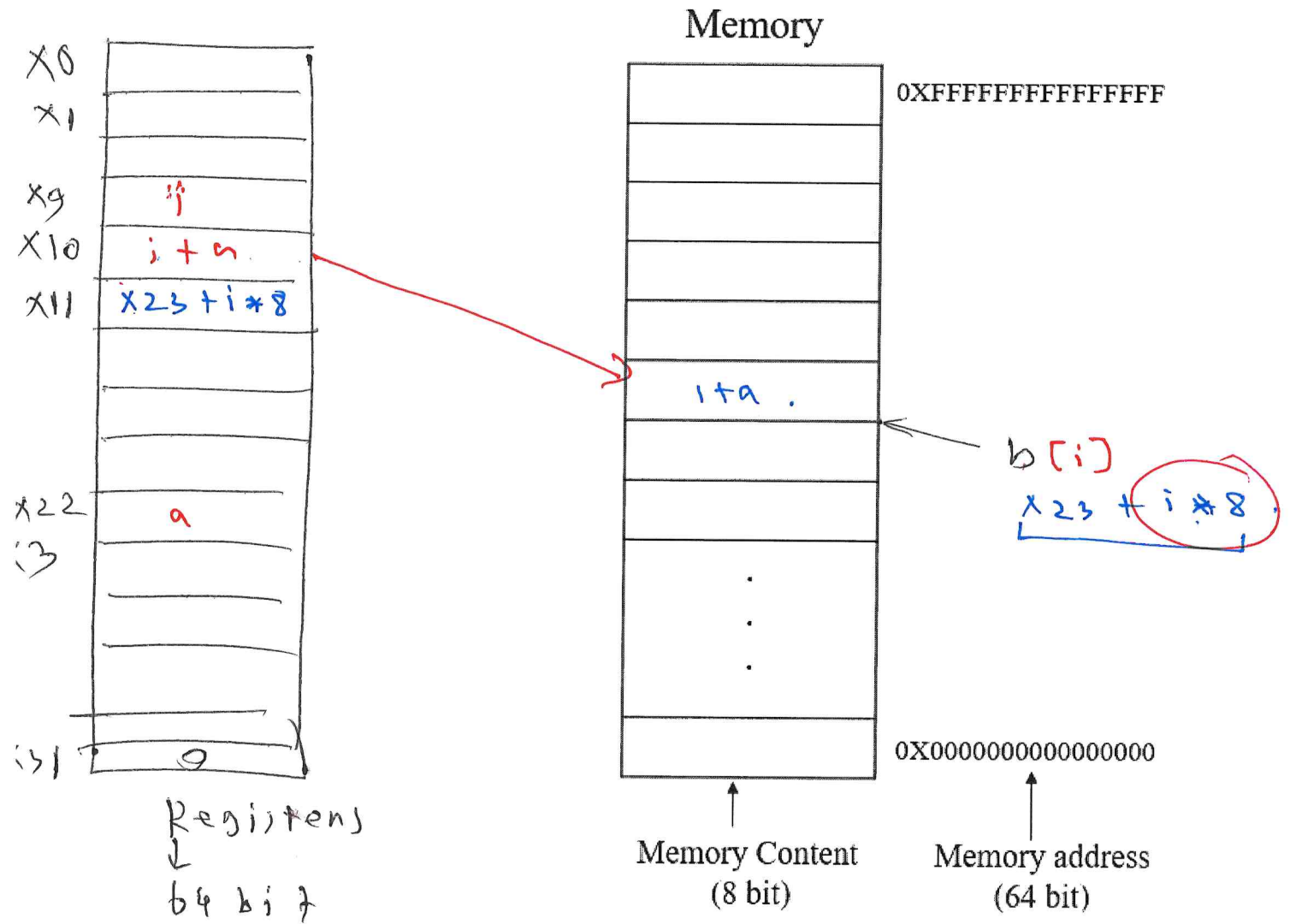
// i++

B Loop

Exit:

22₁₀ = 1 0 1 0

$$b[i] = \overset{x22}{a} + \overset{x9}{i}$$



$$\begin{aligned} \text{Address of } b[i] &= \text{Base address} + i * 8 \\ &= x23 + i * 8 \end{aligned}$$