LSL

LSR – Logical shift Right

- LSR instruction effectively divides the contents of a register by 2^i. For example, the below LSR shifts by 3, which divide the contents of X2 by 2^3 or 8 and place the result on destination register X0.
- Each bit of the register is shifted right, the LSB is removed, and empty bits are filled with zeros. X2 = 24

LSR X0, X2, #3 // $X0 = \frac{X2}{2^3}$

i

8 4 2 ~~1~~

1100 = 12

$X0 = \frac{X2}{2^3} = \frac{24}{8} = 3$

Problem: Assume that X2 = 24 (0X0000000000000018 in hexadecimal). What will be the value of X0 after running the following instruction: LSR X0, X2, #2

LSR X0, X2, #2 // $X0 = \frac{X2}{2^2} = \frac{24}{4} = 6$

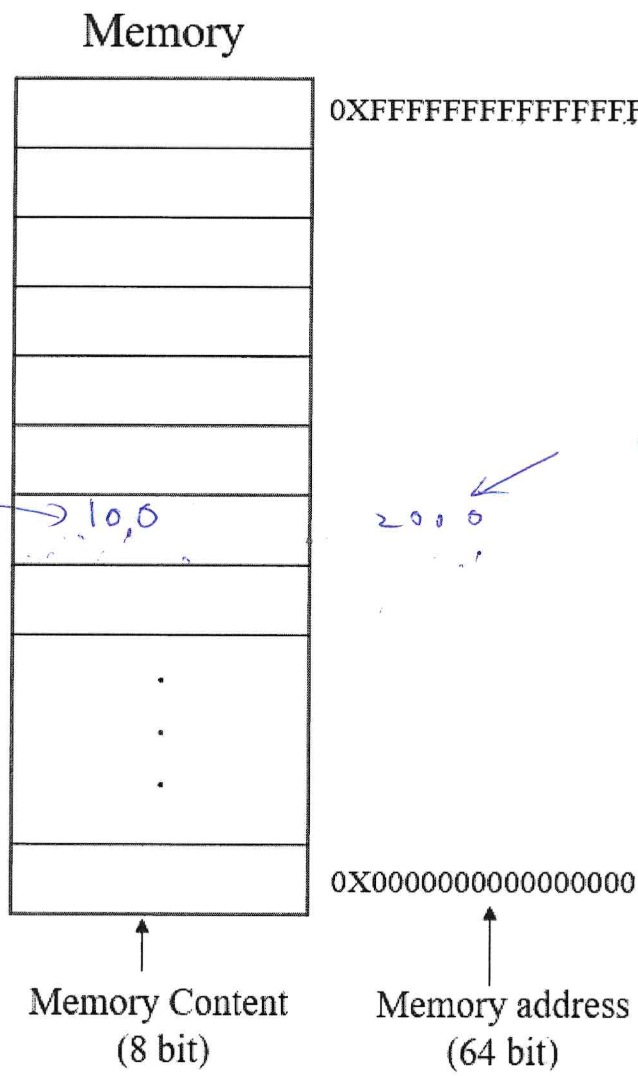
0 0 1 8

X2 0000 0000 0001 100~~0~~ (LSB)

X2 (1st shift) $\xrightarrow{12}$ 0000 0000 0000 110~~0~~

X2 (2nd shift) $\xrightarrow{6}$ 0000 0000 0000 0110

Memory



$2^{64} - 1$

Memory location

Memory content (8 bit)

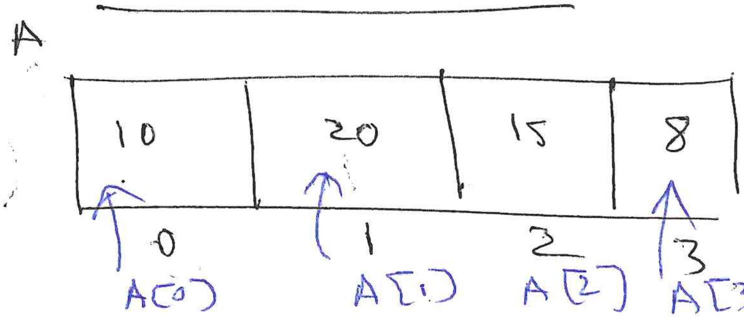
(0)

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

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. (0)

64 bit = 16 Hex digit

64 bit



10 = 0X0000 0000 0000 000A

20 = 0X0000 0000 0000 0014

15 = 0X0000 0000 0000 000F

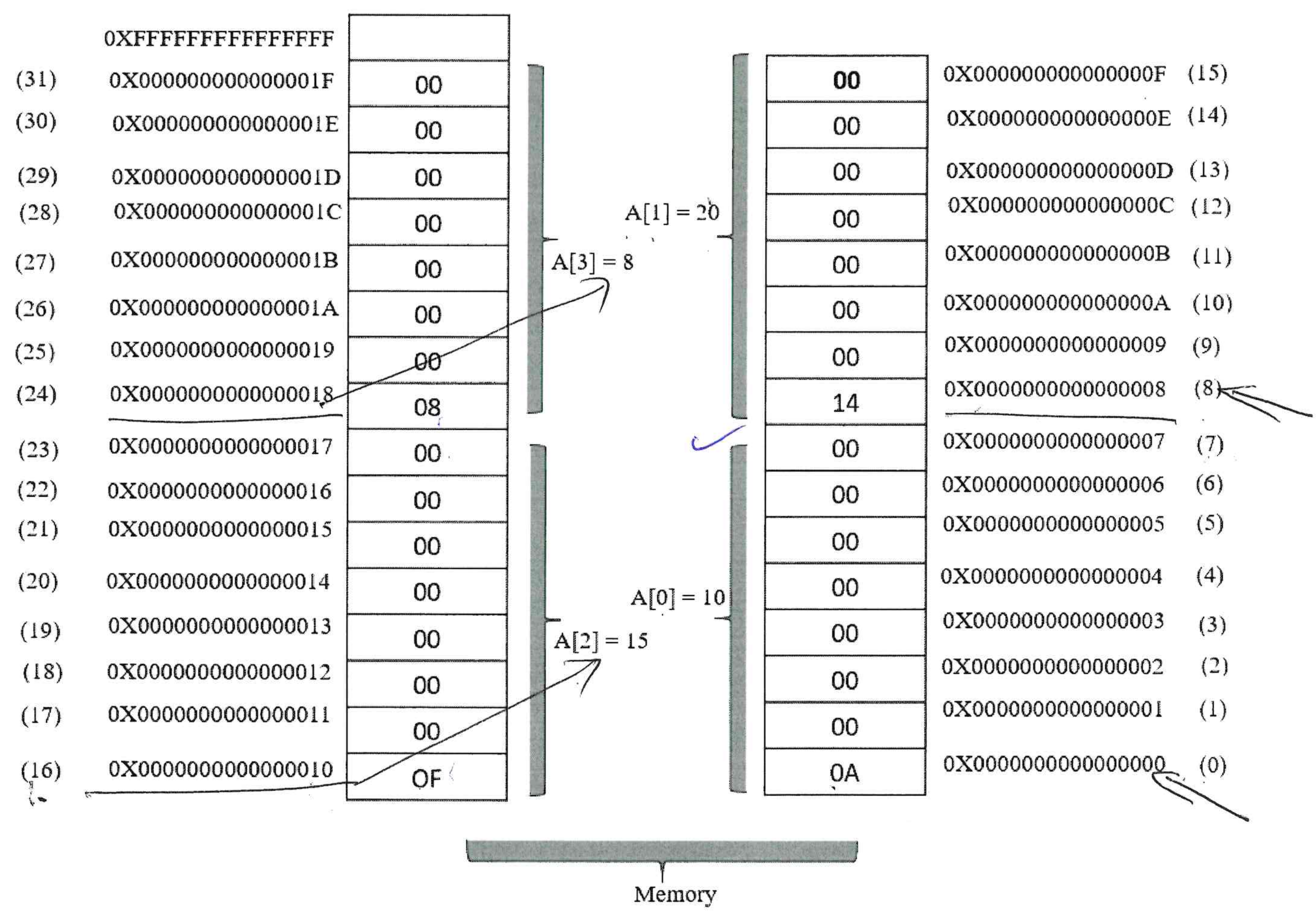
8 = 0X0000 0000 0000 0008

Addres of A[0] = Base address + 0*8 = 0X0000 0000 0000 0000

Address of A[1] = Base addrss + 1*8 = 0X0000 0000 0000 0008

Address of A[2] = Base Address + 2*8 = 0X0000 0000 0000 0010

Address of A[i] = Base address + i*8

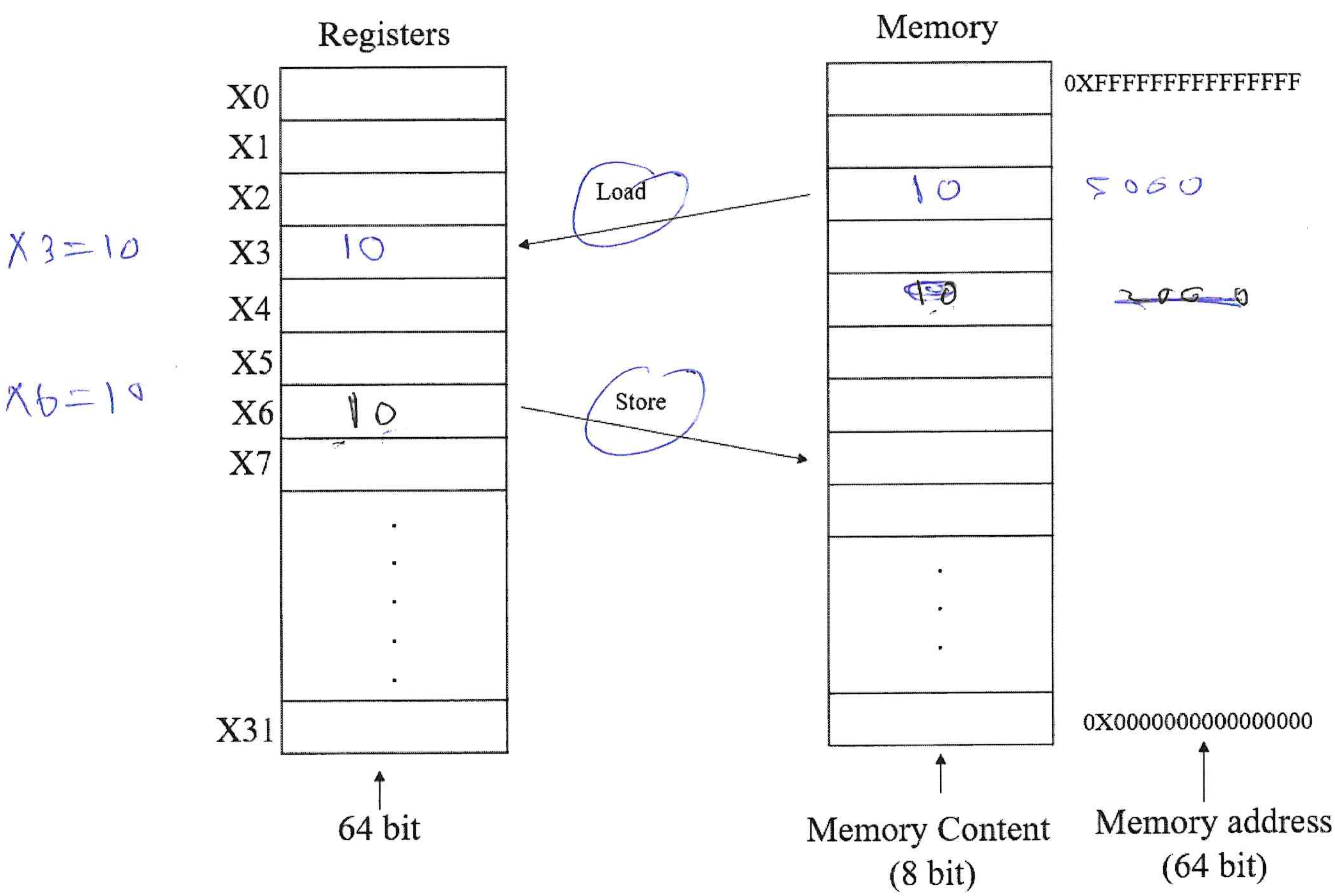


LDUR STUR

Load/Store Instruction

The ARM CPU allows direct access to all locations in the memory, but they are done with specific instructions. Since these instructions either load the register with data from memory or store the data in the register to the memory, the are called load/store instructions.

ADD X0, X3, X6 $\Rightarrow$ X0 = X3 + X6 = 10 + 10 = 20



Load Instruction

- LDUR X2, [X5, #0] instruction copies the content of the memory locations pointed by X5 + 0 into register X2.
- Since the X2 register is a 64-bit wide, it expects a 64-bit operand in the range of 0X0000000000000000 to 0XFFFFFFFFFFFFFFFF.

LDUR X2, [X5, #0]

X5 + 0 =

Memory,

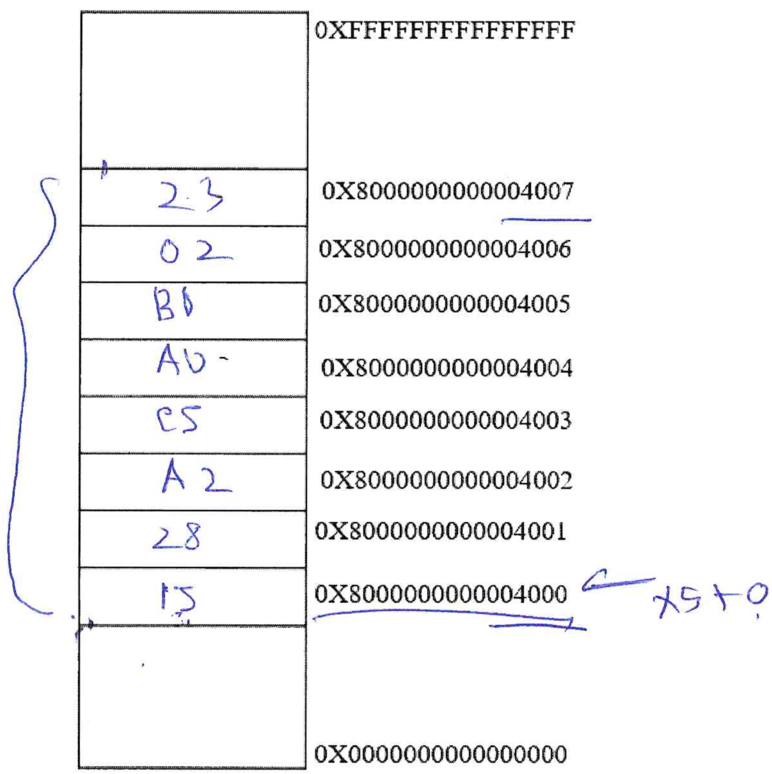
Problem: Assume that X5 = 0X8000000000004000 and locations 0X8000000000004000 through 0X8000000000004007 contain 0X15, 0X28, 0XA2, 0XC5, 0XA0, 0XB1, 0X02, and 0X23, respectively. What will happen to X2 after running the following instruction: LDUR X2, [X5, #0].

X5 + 0 = 0X8100 0000 0000 4000

X5+0

X2 = 2302B0A0C5A22815.

Hex value



Memory

Example: Convert the following C++ code to LEGv8 Assembly code. Assume A is an Array of 10 doublewords, variable g and h are associated with registers X20 and X21, and base address of the array A is in X22.

(1) g = h + A [1];

(2) g = h + A [8];

A → $\boxed{X22}$ = base address

g → X20

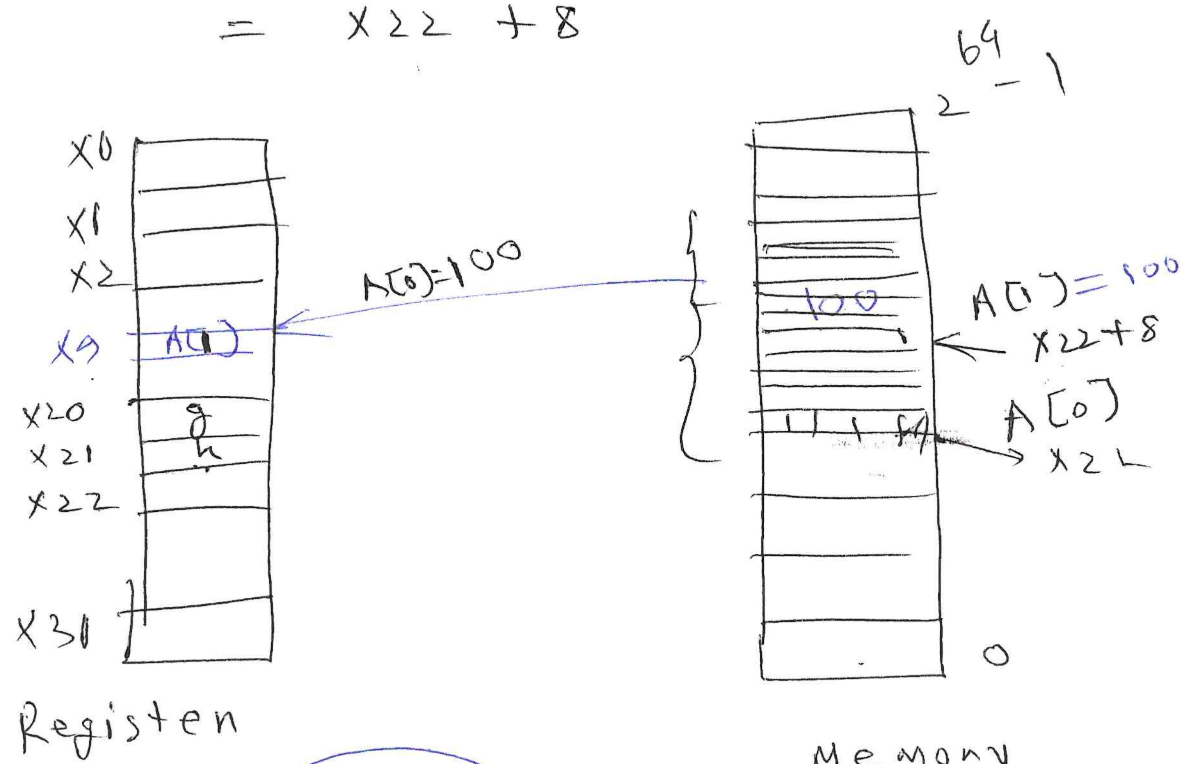
h → X21

1

~~Base~~

Address of A[1] = Base address + 1 * 8

= X22 + 8




```
LDUR  X9, [X22, #8]    // X9 = A[1]
ADD   X20, X9, X21     // g = h + A[1]
```