भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

# Computer Architecture

Assignment -2 - CS2323

Kartik Srinivas - ES20BTECH11015
October 3, 2022

# Index

## 1 Section - 3

The first instruction will load bits 12 through 31 inside the register x3, using the value that is stored , the next instruction will load half-word (in an unsigned manner)
Half word = 16 bits = 4 hexadecimal digits. The lower 4 Hexadecimal digits from byte 0 and byte 1 will be used for storage within the least significant bits of the register x3.
**x3 = 0x000000000000a5a5**
Now if instead we load in a signed manner, the CPU will first check the MSB of the 16 bit number being loaded. This a5a5 has the sign bit of **0xa = 1010** .Therefore there is an **extension** of 1's while loading (which is equivalent to the
**x3 = 0x**$ffffffffffff$**a5a5**
Now loading from an offset of 2 will load **Byte 3 and Byte 4** into the register along with thw usual sign extension
**x3** = **0x**$ffffffffffffa5_{byte-4}a5_{byte-3}$
Now we load the entire double from Byte-0 onwards, we will get the first double word inside the register.
**x3** = **0x39933939a55aa5a5**
Now we load from **Byte 5 onwards**. Since the memory is contiguous, it will start reading into the **next double word** The number stored is then a mix of the higher bytes from the first doubble word and the lower of the second one.
**x3** = **0x**$39933939_{second}39933939_{first}$
Next instruction will just load the eight'th byte(7 have been offset)
**x3** = **0x0000000000000039**
Now even if I use **lb** , it won't make a difference because **0x3 = 0011** and the sign extension will be just of 0's.
**x3** = **0x0000000000000039**
Now , 7'th Byte actually is **0x93 = 1001 0011**, thsi would imply a sign extension when loaded
**x3** = **0xffffffffffffff93**
For code please see [Sri13]

## References

[Sri13] Kartik Srinivas. Cs2323 - computer architecture. https://github.com/kartiksrinivas007/CS2323-CArch.git, 2013.