**SET C- PART B**

**1. How does the hexadecimal value 4F52 get stored in a big-endian computer and a little-endian computer? Convert into binary and show how each byte gets stored.**

**4F52 – 0100111111010010**

**BIG ENDIAN COMPUTER**

|  |  |
| --- | --- |
| **ADDRESS – 1000** | **ADDRESS- 1001** |
| **4F** | **52** |

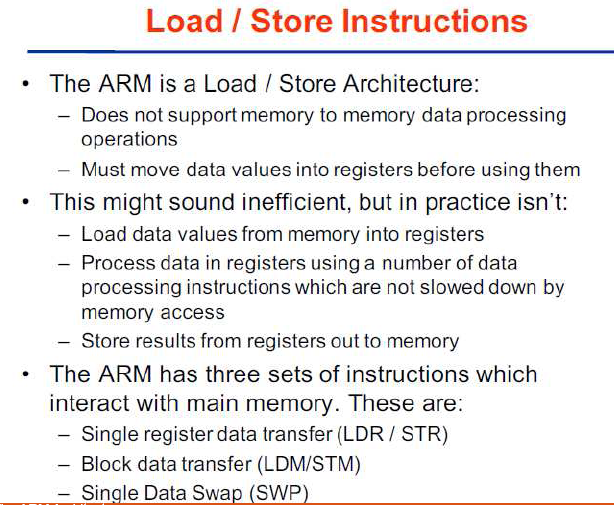
**LITTLE ENDIAN COMPUTER**

|  |  |
| --- | --- |
| **ADDRESS -1000** | **ADDRESS – 1001** |
| **52** | **4F** |

**2. Illustrate memory load and store instructions in ARM**

LDR R2, [R0] @ [R0] - origin address is the value found in R0.

STR R2, [R1] @ [R1] - destination address is the value found in R1.

****

**3. A computer has 128 GB (Gigabytes) of memory. How many bits are needed to address any single byte in memory?**

**A computer has 32KB of memory. Each word in this computer is eight bytes. How many bits are needed to address any single word in memory?**

A. The memory address space is 128 GB, or 237 (27 × 230). This means that we need log2 237, or 37 bits, to address each byte.

B. The memory address space is 32 KB, which means 215. However, each word is eight (23) bytes, which means that we have 212 words. This means that we need log2 212, or 12 bits, to address each word.