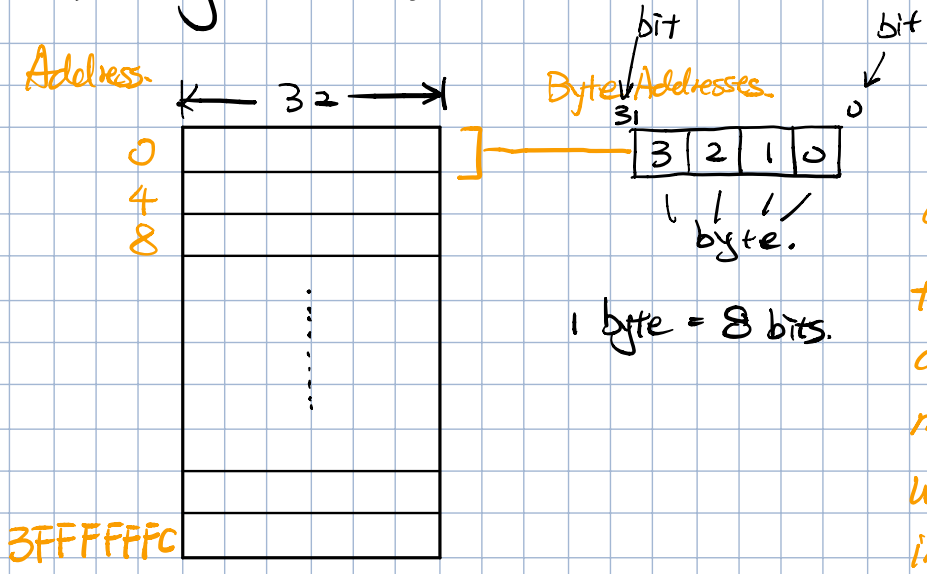
# Memory Architecture.




1 byte = 8 bits.

The least significant byte in a word has the same addr. of the word.

this is known as little Endian addressing. Other processors might use a big Endian. which the most significant byte in a word has the same addr. as the word.

---

-If we run this program on our ARM processor:

```
int main() {
  unsigned int x = 0x76543210;
  char* c_ptr = (char*)&x;
  if (*c_ptr == 0x10)
      (printf "Little Endian");
  else
      (printf "Big Endian");

  return 0;
}
```

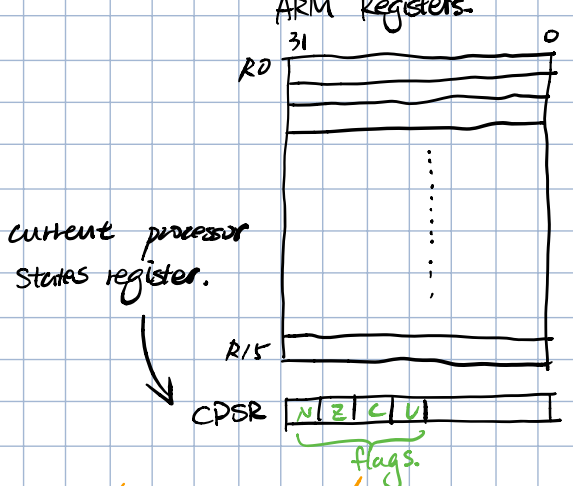
it prints "Little Endian"

---

```
// sum a list of 4 integers.

        .text
        .global  _start

_start: MOV     R0, #4
        MOV     R1, #LIST
```



```
        LDR     R2, [R0]    // get the 1st list element.
        ADD     R1, R0
        LDR     R3, [R1]    // get 2nd list element.
        ADD     R2, R3
        ADD     R1, R0
        LDR     R3, [R1]    // get the 3rd list element.
        ADD     R2, R3
        ADD     R1, R0
        LDR     R3, [R1]    // get the 4th list element.
        ADD     R2, R3      // R2 holds the sum.
END:    mov     R15, #END   // stay here.
LIST:   .word   10, 20, 30, 40
        .end
```