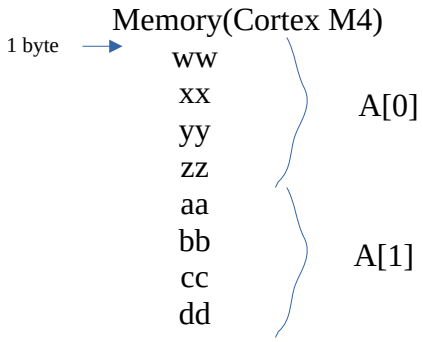
Hints:

- **How to calculate 1 second delay of a system?**
  - Lets the clock frequency of your system = C MHz
  - Delay count= C/desired time duration (for your case, it is 1sec) = C/1= X clock cycles , which means you have to wait X clock cycles for a 1 sec delay in your program.

- **Problem 6:**
  - Declare two registers as two counters C1 and C2 for 1's and 10's place tracking
  - Initially both are zero.
  - Increment 1's place counter
  - Chk whether 1's place reach '10' or not
    - Yes: then increment 10's place counter and handle the 1's place counter accordingly
    - No: Chk whether 10's place reach '10' or not
      - No: introduce delay and display
      - Yes: reset both the counters to 0 means start from beginning "00".

- **Problem 4 and 8:**
- 4 bytes= 32 bits
  - **Problem 4:** d_array DCD 0xaabbccdd, 0xeeffgghh, ……. ; Data array
  - **Problem 8:** d_array DCD 0xaabbcc, 0xeeffgg, ……. ; Data array
- Memory arrangement:

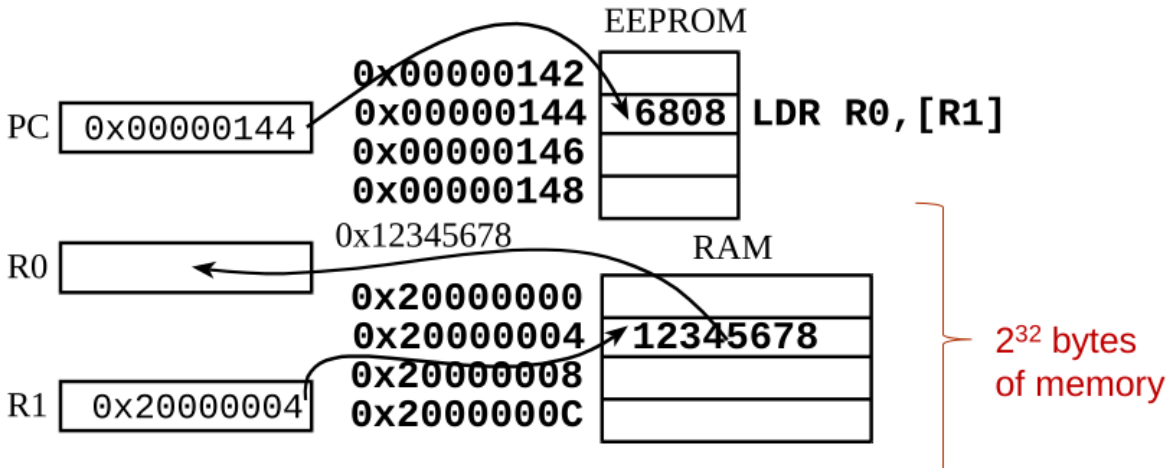


- For byte operation, use LDRB, then you can handle all the bytes separately.
- Separate the bytes using LDRB and also with register indirect addressing mode to traverse through individual bytes [you need to use pre-index and post-index addressing mode] where the incremental value is 1.

- **Reviewing the Addressing Modes of Cortex M4:**

- Memory Operand: ***Register-Indexed*** Addressing
  - A register contains the memory address of *(points to)* the data
  - ***Address*** equivalent to an ***index*** into the array of memory bytes

```
LDR   R0,[R1]        ; R0= value pointed to by R1
```



Example

```
STR r1, [r0, #4]!      ;pre-increment
; r0 = 0x20008000,   r1=0x76543210
```

r0 before store

0x20008000

r0 after store

0x20008004

Memory Address	Memory Data
0x20008007	0x76
0x20008006	0x54
0x20008005	0x32
0x20008004	0x10
0x20008003	0x00
0x20008002	0x00
0x20008001	0x00
0x20008000	0x00

Example

```
STR r1, [r0], #4   ;post-increment
; r0 = 0x20008000,  r1=0x76543210
```

r0 before store

0x20008000

r0 after store

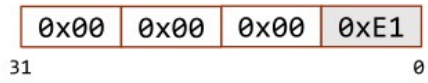
0x20008004

Memory Address	Memory Data
0x20008007	0x00
0x20008006	0x00
0x20008005	0x00
0x20008004	0x00
0x20008003	0x76
0x20008002	0x54
0x20008001	0x32
0x20008000	0x10

Load a Byte, Half-word, Word

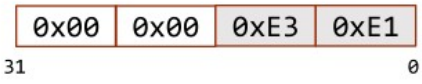
Load a Byte

```
LDRB r1, [r0]
```



Load a Halfword

```
LDRH r1, [r0]
```



Load a Word

```
LDR r1, [r0]
```