Chapter 3 · Section 3.5 — Exercises (Mazidi)

Chapter 3 · Section 3.5 — Exercises (Mazidi)

> Problems are paraphrased to respect copyright. Results are shown in **8-bit hex** where appropriate.

---

15) Convert `0x76` (packed BCD) to ASCII digits; place ASCII codes in R1 and R2

Approach

- Extract **tens** = high nibble `(value >> 4)` and **ones** = low nibble `(value & 0xF)`.
- Convert each digit to ASCII by **adding 0x30**.

Solution

```
        AREA    |.text|, CODE, READONLY
        EXPORT  _start
        THUMB
_start:
        MOVS    r0, #0x76         ; packed BCD = 0x76 (digits 7 and 6)

        LSRS    r1, r0, #4        ; r1 = 0x07  (tens)
        ANDS    r2, r0, #0x0F     ; r2 = 0x06  (ones)

        ADDS    r1, r1, #0x30     ; ASCII '7' = 0x37
        ADDS    r2, r2, #0x30     ; ASCII '6' = 0x36

        B       .
        END
```

Result: `R1 = 0x37` (ASCII '7'), `R2 = 0x36` (ASCII '6').

---

16) Keyboard provides ASCII `0x33` ('3') and `0x32` ('2'). Convert them to packed BCD and store in R2

Approach

- Convert ASCII to numeric: **subtract 0x30** from each.
- Pack: `(tens << 4) | ones`.

Solution

```
        AREA    |.text|, CODE, READONLY
        EXPORT  _start
        THUMB
_start:
        MOVS    r0, #0x33         ; ASCII '3'
        MOVS    r1, #0x32         ; ASCII '2'

        SUBS    r0, r0, #0x30     ; r0 = 3
        SUBS    r1, r1, #0x30     ; r1 = 2

        LSLS    r0, r0, #4        ; r0 = 0x30 (tens in high nibble)
        ORR     r2, r0, r1        ; r2 = 0x32 (packed BCD)

        B       .
        END
```

Result: `R2 = 0x32` (packed BCD “3 2”).

Notes for learners

- **ASCII digit ↔□ numeric digit:** `digit_ascii = digit + 0x30` and `digit = digit_ascii - 0x30`.
- **Packed BCD** stores two 4-bit digits per byte; useful when printing/reading decimal without full division.