

## Chapter 3 · Section 3.2 — Exercises (Mazidi)

## Chapter 3 · Section 3.2 — Exercises (Mazidi)

Problems are paraphrased to respect copyright. Assume **independent** sub-questions unless noted. Initial registers for Q3: **R0 = 0xF000, R1 = 0x3456, R2 = 0xE390**.

### 3) Perform each operation; give the result and the destination register.

- (a) `AND R3,R2,R0` → `R3 = 0xE390 AND 0xF000 = 0xE000`
- (b) `ORR R3,R2,R1` → `R3 = 0xE390 OR 0x3456 = 0xF7D6`
- (c) `EOR R0,R0,#0x76` → `R0 = 0xF000 XOR 0x0076 = 0xF076`
- (d) `AND R3,R2,R2` → `R3 = 0xE390 AND 0xE390 = 0xE390`
- (e) `EOR R0,R0,R0` → `R0 = 0x00000000` (XOR with itself clears)
- (f) `ORR R3,R0,R2` → `R3 = 0xF000 OR 0xE390 = 0xF390`
- (g) `AND R3,R0,#0xFF` → `R3 = 0xF000 AND 0x00FF = 0x00`
- (h) `ORR R3,R0,#0x99` → `R3 = 0xF000 OR 0x0099 = 0xF099`
- (i) `EOR R3,R1,R0` → `R3 = 0x3456 XOR 0xF000 = 0xC456`
- (j) `EOR R3,R1,R1` → `R3 = 0x00000000`

### 4) Value in R2 after executing:

```
MOV    R0,#0xF0
MOV    R1,#0x55
BIC    R2,R1,R0
```

**Answer:** `R2 = R1 AND (~R0) = 0x55 AND 0x0F = 0x05` → **R2 = 0x00000005**.

### 5) Value in R2 after executing:

```
LDR    R1,=0x55555555
MVN    R0,#0
EOR    R2,R1,R0
```

**Answer:** `R0 = ~0 = 0xFFFFFFFF`; `R2 = R1 XOR R0 = ~R1 = 0xAAAAAAAA` → **R2 = 0xAAAAAAAA**.

## Notes for learners

- **BIC Rd,Rn,Op2:** `Rd = Rn AND NOT Op2`.
- **MVN Rd,Op2:** `Rd = NOT Op2`.
- **EOR with itself clears a register; AND with itself preserves it; ORR with anything sets the union of bits.**