Ερώτημα 46

Γιατί πρέπει να προγραμματίσουμε τους καταχωρητές SCU_PCGR1 και SCU_PRR1 με τις τιμές 0x00220000;

Γιατί η τιμή 0x00220000, ενεργοποιεί το ρολόι (clock enable), για τα περιφερειακά GPIO3 και GPIO7 μέσω του SCU_PCGR1, και αφαιρεί το reset (reset release), για τα ίδια περιφερειακά μέσω του SCU_PRR1.

Ερώτημα 47

Πόσους καταχωρητές για το GPIO Port 3 έχουμε δηλώσει; Γιατί δεν τους έχουμε δηλώσει όλους;

Στον κώδικα που αναλύσαμε, για τον GPIO3 (Port 3) έχει δηλωθεί μόνο ένας καταχωρητής, ο GPIO3_DATA_ADDR EQU APB0_NBUF_BASE + 0x00009000.
Αυτός είναι ο καταχωρητής δεδομένων (data register), ο οποίος χρησιμοποιείται για ανάγνωση της κατάστασης των εισόδων (π.χ. κουμπί), ή σε άλλες περιπτώσεις για εγγραφή εξόδων.
Δηλώσαμε μόνο αυτόν, γιατί επαρκεί για τη λειτουργία που θέλουμε, καθώς το κουμπί είναι συνδεδεμένο στην είσοδο GPIO3.5. Δεν χρειαζόμαστε αλλαγή mode, κατεύθυνσης, ή άλλες λειτουργίες και αρκεί να διαβάσουμε την τιμή του P3.5. Καλή τακτική είναι να μην δηλώνουμε όλους τους πιθανούς καταχωρητές του GPIO3 (direction, control, interrupt, lock, κ.λπ.), αν δεν τους χρησιμοποιούμε. Αυτό μειώνει το μέγεθος του κώδικα και την πολυπλοκότητα. Γενικά, χρησιμοποιούμε μόνο ό,τι είναι απολύτως απαραίτητο για τη λειτουργία που υλοποιούμε.

Ερώτημα 48

Εάν το σύστημα τρέχει στα 25MHz πόσο χρόνο θα διαρκέσουν οι 10 NOP εντολές;

Εφόσον κάθε NOP καταναλώνει 1 κύκλο και το σύστημα τρέχει στα 25 MHz, οι 10 NOP εντολές διαρκούν συνολικά 400 νανοσεκόντ ή 0.4 μικροσεκόντ.
Χρόνος 1 κύκλου (T):
T = 1 / f
T = 1 / 25.000.000 = 40 ns (νανοσεκόντ)
Χρόνος για 10 NOP:
10 εντολές × 1 κύκλος/εντολή × 40 ns = 400 ns = 0.4 μs (μικροσεκόντ)

Ερώτημα 49

Ποια θα περιμέναμε να είναι τα αποτελέσματα των τριών παραπάνω εντολών;

Οι τρεις εντολές έχουν ως αποτέλεσμα να ενεργοποιηθεί η παροχή ρολογιού (clock signal) στα GPIO3 και GPIO7, γράφοντας την κατάλληλη τιμή στον καταχωρητή SCU_PCGR1. Αυτό είναι **απαραίτητο βήμα** πριν χρησιμοποιήσουμε τα περιφερειακά, καθώς χωρίς ενεργό ρολόι δεν ανταποκρίνονται σε καμία εντολή.

Ερώτημα 50
Ποιά θα περιμέναμε να είναι τα αποτελέσματα των δύο παραπάνω εντολών; Τα δεδομένα του R0 δεν έχουν μεταβληθεί από την πρώτη εντολή: LDR R0, =SCU_BASE

Οι GPIO3 και GPIO7 βγαίνουν από την κατάσταση reset και πλέον είναι έτοιμοι να λειτουργήσουν — δηλαδή μπορούν να δεχτούν ρυθμίσεις, να διαβάσουν/γράψουν δεδομένα, κ.λπ.
Οι δύο εντολές έχουν ως αποτέλεσμα τη μετάβαση των GPIO3 και GPIO7 από reset σε ενεργή λειτουργία. Χωρίς αυτή την εγγραφή, τα περιφερειακά θα παρέμεναν ανενεργά — ακόμα κι αν είχαν clock. Το R0 παραμένει αμετάβλητο, καθώς εξακολουθεί να δείχνει στη βάση του SCU (0x5C002000), διευκολύνοντας τις εγγραφές μέσω offsets.

Ερώτημα 51
Συνεχίστε τον Lab6_GPIOs_assembly.s κώδικα ώστε το πάτημα του πλήκτρου INT6 να ανάβει το LED 1 (P 7.1 στο σχηματικό).

```
1. Check_INT6_Button
2.      LDR     R0, =GPIO3_DATA_ADDR        ; Διεύθυνση του GPIO3
3.      LDRB    R0, [R0, #0x080]            ; Ανάγνωση κατάστασης (πιθανόν P3.6)
4.      TST     R0, #0x40                   ; Έλεγχος του bit 6 (0x40 = 0b01000000)
5.      BEQ     Led1_ON                     ; Αν είναι 0 (πατημένο) → άναψε LED1
6.      B       Led1_OFF                    ; Αλλιώς → σβήσε LED1
7.
```

Νέα υποπρογράμματα: Led1_ON και Led1_OFF

```
 1. Led1_ON
 2.      LDR     R0, =GPIO7_DATA_ADDR
 3.      LDR     R1, =0x02                   ; Bit 1 HIGH → άναψε LED1 (P7.1)
 4.      STR     R1, [R0, #0x004]
 5.      B       Check_INT6_Button           ; Επιστροφή στον έλεγχο
 6.
 7. Led1_OFF
 8.      LDR     R0, =GPIO7_DATA_ADDR
 9.      LDR     R1, =0x00                   ; Bit 1 LOW → σβήσε LED1
10.      STR     R1, [R0, #0x004]
11.      B       Check_INT6_Button           ; Επιστροφή στον έλεγχο
12.
```