

Ερώτημα 52: SCU_PCGR0

SCU_PCGR0 σημαίνει: *System Control Unit Peripheral Clock Gating Register 0*

Τιμή: Συνήθως μια τιμή όπως 0x000003FC ή παρόμοια (θα εξαρτηθεί από το ποιες περιφερειακές μονάδες χρησιμοποιούνται).

Εξήγηση:

Ο καταχωρητής αυτός ενεργοποιεί ή απενεργοποιεί την παροχή ρολογιού (clock gating) σε περιφερειακές μονάδες του συστήματος.

Για κάθε bit:

- **Bit = 1** → το αντίστοιχο περιφερειακό **λαμβάνει ρολόι** (είναι ενεργό)
- **Bit = 0** → **δεν λαμβάνει ρολόι** (είναι απενεργοποιημένο για εξοικονόμηση ενέργειας)

Παράδειγμα:

- Bit 2: UART0
- Bit 3: UART1
- Bit 4: Timer0
- Bit 5: Timer1
- Bit 6: Timer2

Άρα:

```
1. SCU_PCGR0 = 0x0000007C; // Ενεργοποιεί Timer0, Timer1, Timer2, UART1, UART0
```

Ερώτημα 53: SCU_PRR0

SCU_PRR0 σημαίνει: *System Control Unit Peripheral Reset Register 0*

Τιμή: Μια τυπική τιμή αρχικοποίησης μπορεί να είναι 0x00000000, και κατόπιν γίνεται εγγραφή 1 στα bits που πρέπει να **βγουν από reset**.

Εξήγηση:

Αυτός ο καταχωρητής χρησιμοποιείται για να κάνουμε **reset σε περιφερειακά** (κάθε bit αντιστοιχεί σε μία περιφερειακή μονάδα):

- **Bit = 1** → το αντίστοιχο περιφερειακό τίθεται σε **reset**
- **Bit = 0** → λειτουργεί κανονικά

Αν θέλουμε να απενεργοποιήσουμε το reset (δηλαδή να ενεργοποιήσουμε τη λειτουργία) ενός UART και Timer:

```
1. SCU_PRR0 &= ~( (1 << 2) | (1 << 4) ); // Βγάζει από reset το UART0 και Timer0
```

Ερώτημα 54: SCU_PRR1

SCU_PRR1 = *Peripheral Reset Register 1* (2ο μέρος του SCU_PRR)

Τιμή: Π.χ. 0x00000000 ή ~(1 << X)

Εξήγηση:

Ελέγχει το **reset σε περιφερειακά** που δεν περιλαμβάνονται στο SCU_PRR0, π.χ.:

- CAN, I²C, SPI, ADC, κ.λπ.

- Η εγγραφή 1 σε bit θέτει το περιφερειακό σε reset.
- Συνήθως, ο κώδικας εκτελεί AND με το συμπλήρωμα του bit ($\sim(1 \ll n)$) για να **αφαιρέσει** το reset σε κάποιο περιφερειακό που θα χρησιμοποιηθεί.

Παράδειγμα:

```
1. SCU_PRR1 &= ~(1 << 3); // Βγάζει το SPI από reset (αν το bit 3 είναι για SPI)
```

Ερώτημα 55: SCU_WKUPSEL

SCU_WKUPSEL = *Wakeup Pin Selection Register*

Τιμή: Π.χ. 0x00000002

Εξήγηση:

Ορίζει **ποιες είσοδοι (pins)** μπορούν να χρησιμοποιηθούν για **έξοδο από sleep mode** (wake-up event).

Κάθε bit αντιστοιχεί σε έναν wake-up pin:

- **Bit = 1** → ενεργοποίηση wake-up για το αντίστοιχο pin
- **Bit = 0** → αγνοείται το αντίστοιχο pin

Παράδειγμα:

```
1. SCU_WKUPSEL = 0x00000002; // Ενεργοποιεί wake-up μέσω του pin 1
```

Ερώτημα 56: VIC_INTER

VIC_INTER = *Vector Interrupt Controller Interrupt Enable Register*

Τιμή: Π.χ. 0x00000004

Εξήγηση:

Ενεργοποιεί τις **διακοπές (interrupts)** για συγκεκριμένες πηγές:

- **Bit = 1** → Η αντίστοιχη διακοπή είναι ενεργή
- **Bit = 0** → Η διακοπή αγνοείται

Κάθε bit αναφέρεται σε μία διακοπή, π.χ.:

- Bit 2 → Timer 0
- Bit 4 → UART0
- Bit 7 → External IRQ7

Παράδειγμα:

```
1. VIC_INTER = (1 << 2); // Ενεργοποιεί διακοπή από Timer0
```

Ερώτημα 57: VIC1_VA10R

VIC1_VA10R = *VIC Vector Address Register 10 (interrupt slot 10)*

Τιμή: Π.χ. 0x00008020

Εξήγηση:

Περιέχει τη **διεύθυνση της ρουτίνας εξυπηρέτησης διακοπής (ISR)** για το **slot 10** του πίνακα διακοπών.

Δηλαδή:

- Όταν ενεργοποιηθεί διακοπή που έχει προτεραιότητα και είναι στο slot 10, τότε το πρόγραμμα πηγαίνει στη διεύθυνση 0x00008020.

Παράδειγμα:

```
1. VIC1_VA10R = (unsigned int)&Timer0_IRQHandler;
```

Όπου Timer0_IRQHandler είναι η συνάρτηση χειρισμού διακοπής.

Ερώτημα 58: VIC1_VC10R

VIC1_VC10R = VIC Vector Control Register for Slot 10

Τιμή (ενδεικτική): π.χ. 0x00000024

Εξήγηση:

Αυτός ο καταχωρητής:

- Ενεργοποιεί ή απενεργοποιεί τη διακοπή για το συγκεκριμένο vector slot (εδώ slot 10)
- Καθορίζει **ποια πηγή διακοπής** θα συνδεθεί με το συγκεκριμένο vector slot

Η μορφή του:

- **Bit 5–0:** Κωδικός πηγής διακοπής (π.χ. 0x04 = Timer0)
- **Bit 10:** Ενεργοποίηση του slot (bit = 1)

Παράδειγμα:

```
1. VIC1_VC10R = 0x00000024;
```

- 0x04 → Timer0
- 0x20 → Ενεργοποίηση slot (bit 5)
- Άρα, ενεργοποιεί το slot 10 για τη διακοπή του Timer0

Ερώτημα 59: SCU_GPIOOUT7

SCU_GPIOOUT7 = SCU Output Control Register for GPIO Port 7

Τιμή (ενδεικτική): π.χ. 0xAAAAAAAA

Εξήγηση:

Ορίζει τον **τύπο εξόδου** για κάθε pin του GPIO7. Κάθε 2 bits ορίζουν μία έξοδο:

- 00 → Push-pull
- 01 → Open-drain
- 10 → Reserved
- 11 → Not used / undefined

Η τιμή 0xAAAAAAAA σημαίνει:

- Εναλλάσσει τα bits 10 (σε δεκαεξαδικό A = 1010)

- Άρα: Open-drain για όλα τα pins (όπου ισχύει)

Παράδειγμα:

```
1. SCU_GPIOOUT7 = 0x00000000; // Όλες οι έξοδοι push-pull
2. SCU_GPIOOUT7 = 0xAAAAAAAA; // Εναλλακτικές εξόδους open-drain
```

Ερώτημα 60: GPIO7_DIR

GPIO7_DIR = *Direction Register for GPIO Port 7*

Τιμή (ενδεικτική): π.χ. 0x0000000F

Εξήγηση:

Ορίζει την **κατεύθυνση** (input/output) κάθε pin του GPIO7:

- **Bit = 1** → Output
- **Bit = 0** → Input

Η τιμή 0x0F σημαίνει:

- Τα **τέσσερα λιγότερο σημαντικά bits** (pins 0–3) είναι **έξοδοι**
- Τα υπόλοιπα pins είναι **είσοδοι**

Παράδειγμα:

```
1. GPIO7_DIR = 0x0000000F; // GPIO7.0 έως GPIO7.3 ως έξοδοι
```

Ερώτημα 61: WIU_CTRL

WIU_CTRL = *Wake-up Interrupt Unit Control Register*

Τιμή (ενδεικτική): π.χ. 0x00000001

Εξήγηση:

Ρυθμίζει τον **έλεγχο της Wake-up Interrupt Unit (WIU)**, δηλαδή:

- Ενεργοποίηση της μονάδας
- Επιλογή τρόπου ενεργοποίησης (edge/level trigger)
- Συνήθως, το bit 0 χρησιμοποιείται για ενεργοποίηση:
 - **Bit 0 = 1** → Ενεργοποίηση του WIU
 - Άλλα bits μπορεί να ορίζουν αν θα ενεργοποιείται σε positive/negative edge

Παράδειγμα:

```
1. WIU_CTRL = 0x00000001; // Ενεργοποίηση της Wake-up Interrupt Unit
```

Ερώτημα 62: WIU_MASK (γνωστός και ως WIU_MR)

WIU_MASK / WIU_MR = *Wakeup Interrupt Unit Mask Register*

Τιμή (ενδεικτική): π.χ. 0x00000004

Εξήγηση:

- Ενεργοποιεί ή **“ξεμασκάρει”** τα wake-up pins που επιτρέπεται να προκαλέσουν διακοπή.
- Κάθε **bit αντιστοιχεί σε μία γραμμή wake-up εισόδου (WKP)**.

- **Bit = 1** → το αντίστοιχο pin μπορεί να προκαλέσει διακοπή.
- **Bit = 0** → το pin **αγνοείται** από τη WIU.

Παράδειγμα:

```
1. WIU_MR = 0x00000004;
```

- Μόνο η **γραμμή WKP2** (bit 2) είναι ενεργή για wake-up.

Έτσι, αν ένα pin αλλάξει κατάσταση, μόνο το WKP2 θα προκαλέσει αφύπνιση.

Ερώτημα 63: WIU_TRIG (γνωστός και ως WIU_TR)

WIU_TRIG / WIU_TR = *Wakeup Interrupt Unit Trigger Register*

Τιμή (ενδεικτική): π.χ. 0x00000004

Εξήγηση:

- Καθορίζει τη **συνθήκη ενεργοποίησης (trigger type)** για κάθε wake-up είσοδο:
 - **Bit = 0** → Ενεργοποίηση σε **αρνητική ακμή** (falling edge)
 - **Bit = 1** → Ενεργοποίηση σε **θετική ακμή** (rising edge)

Παράδειγμα:

```
1. WIU_TR = 0x00000004;
```

- Η **γραμμή WKP2** (bit 2) ενεργοποιεί wake-up σε **θετική ακμή** (rising edge)
- Όλες οι υπόλοιπες (με bit 0) ενεργοποιούνται σε αρνητική ακμή (αν είναι ξεμασκαρισμένες)

Συνδυαστικό Παράδειγμα

Για να ενεργοποιηθεί wake-up:

1. Το bit του pin πρέπει να είναι **1 στο WIU_MR**
2. Και η **ακμή** πρέπει να συμφωνεί με το WIU_TR

```
1. WIU_MR = 0x00000004; // Ενεργό μόνο το WKP2
2. WIU_TR = 0x00000004; // Rising edge trigger για WKP2
3. WIU_CTRL = 0x00000001; // Ενεργοποίηση της WIU
```