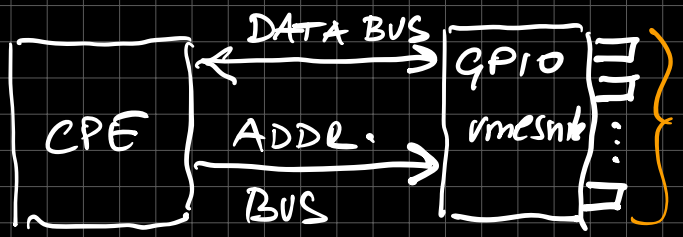
Zadnji:

→ V/I naprave so pomnilniško preslikane

↳ CPE so vidne kot 1 ali več pom. beseda

Prva naprava na vajah: GPIO vmesnik

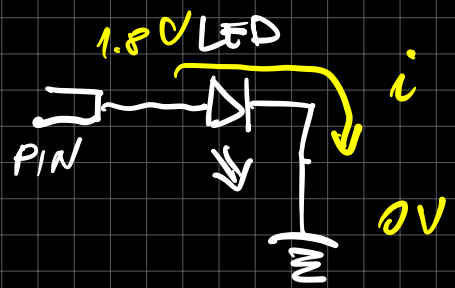
↳ General Purpose I/O



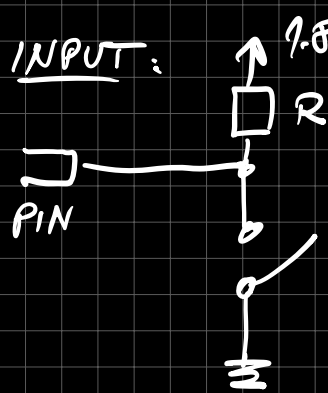
Podatek, ki ga CPE zapiše v GPIO vmesnik se pokaže na Pinih

"0" (0V) "1" (1.8V)

OUTPUT:



INPUT:

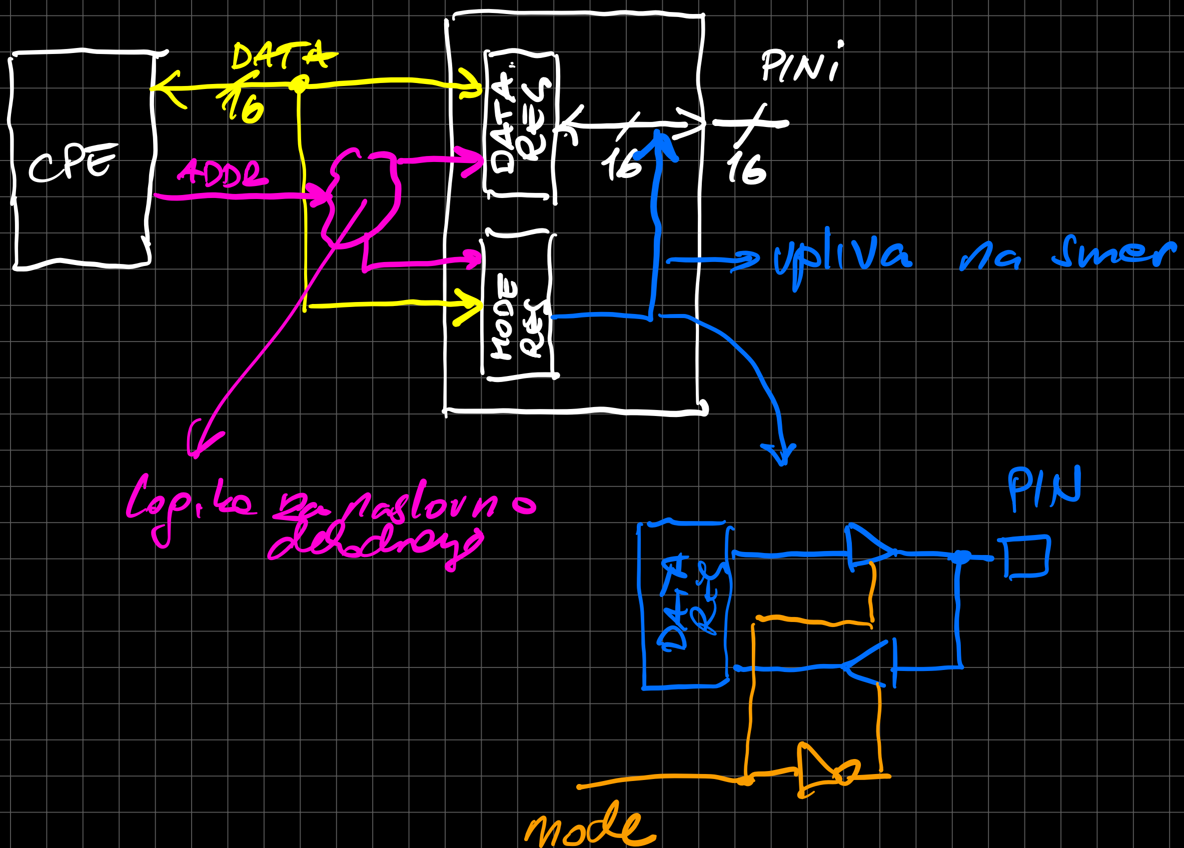


GPIO potrebujejo vsaj dve pom. besedi:

↳ 2 registra

PODATKOVNI REGISTER (DATA REG)
= preslikuje logične nivoje med pod. vodilom in pini

(MODE REG)
REGISTER NAČINA DELOVANJA
↳ določa, ali je posamezni pin vhodni ali izhodni gledano iz CPE



0x4002 2800 - 0x4002 2BFF	GPIOK
0x4002 2400 - 0x4002 27FF	GPIOJ
0x4002 2000 - 0x4002 23FF	GPIOI
0x4002 1C00 - 0x4002 1FFF	GPIOH
0x4002 1800 - 0x4002 1BFF	GPIOG
0x4002 1400 - 0x4002 17FF	GPIOF
0x4002 1000 - 0x4002 13FF	GPIOE
0x4002 0C00 - 0x4002 0FFF	GPIOD
0x4002 0800 - 0x4002 0BFF	GPIOC
0x4002 0400 - 0x4002 07FF	GPIOB
0x4002 0000 - 0x4002 03FF	GPIOA

```
/*!< AHB1 peripherals */
#define GPIOA_BASE            (AHB1PERIPH_BASE + 0x0000UL)
#define GPIOB_BASE            (AHB1PERIPH_BASE + 0x0400UL)
#define GPIOC_BASE            (AHB1PERIPH_BASE + 0x0800UL)
#define GPIOD_BASE            (AHB1PERIPH_BASE + 0x0C00UL)
#define GPIOE_BASE            (AHB1PERIPH_BASE + 0x1000UL)
#define GPIOF_BASE            (AHB1PERIPH_BASE + 0x1400UL)
#define GPIOG_BASE            (AHB1PERIPH_BASE + 0x1800UL)
#define GPIOH_BASE            (AHB1PERIPH_BASE + 0x1C00UL)
#define GPIOI_BASE            (AHB1PERIPH_BASE + 0x2000UL)
```

```
/*!< Peripheral memory map */
#define APB1PERIPH_BASE       PERIPH_BASE
#define APB2PERIPH_BASE       (PERIPH_BASE + 0x00010000UL)
#define AHB1PERIPH_BASE       (PERIPH_BASE + 0x00020000UL)
#define AHB2PERIPH_BASE       (PERIPH_BASE + 0x10000000UL)
```

```
#define PERIPH_BASE           0x40000000UL
```

0x40000000 + 0x00020000 + 0x0800 =
0x40020800 → GPIOC

```c
unsigned int *pNaslovGPIOC = GPIOC_BASE;

*pNaslovGPIOC = 0x5555;
```

MODER: 0x40020800 + 0x00

IDR : 0x40020800 + 0x10

ODR : 0x40020800 + 0x14

Kako pa V/I naprave komunicirajo s CPE?

↙

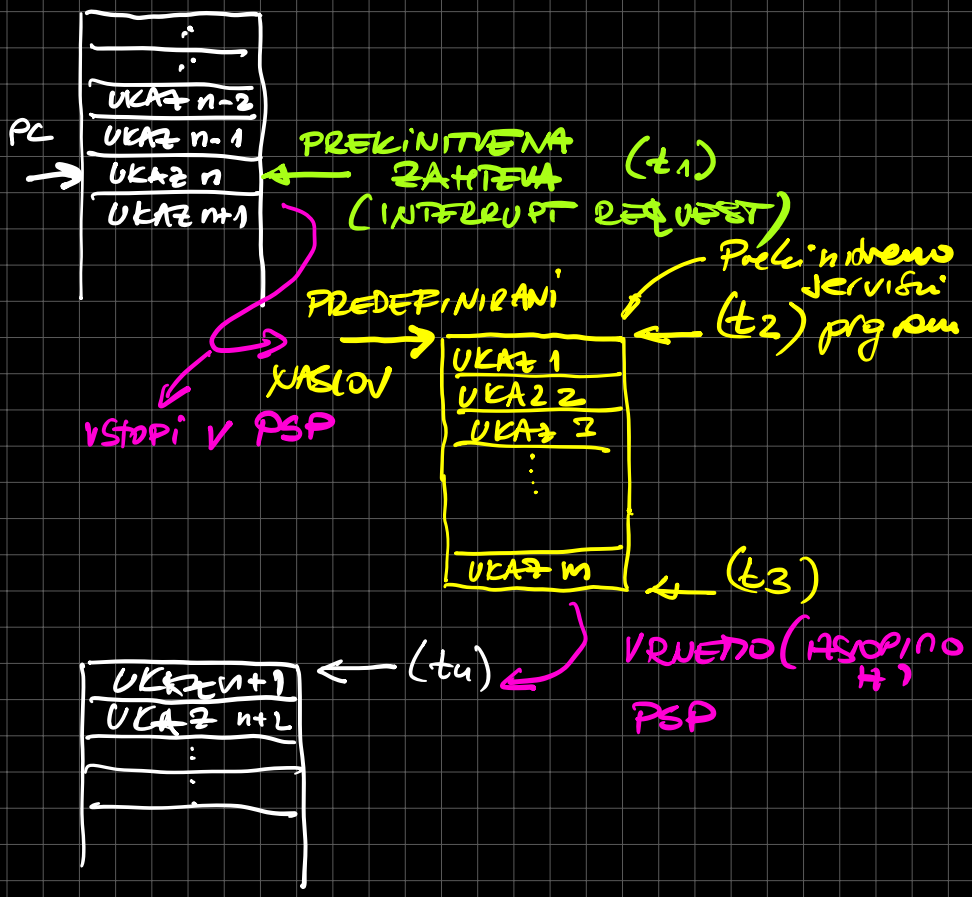
PROBLEMI:

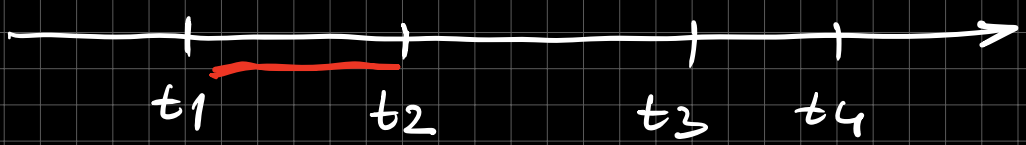
1. V/I naprave [illegible] procesor ali mnogo LOAD/STORE ukazov
2. CPE ni v naslovnem prostoru naprav
   → CPE ni naslovljiv za naprave

⇓

TA NAČIN KOMUNICIRANJE CPE → V/I napravi NI MOGOČ V DRUGO SMER

PREKINITVE





PREKINITVENA LATENCA / ZAKASNITEV
↳ MORA BITI ČIM KRAJŠA

ODZIV CPE NA PREKINITVENO ZAHTEVO

1. ZAKLJUČI UKAZE, KI SE IZVAJAJO IN SPREMINJAJO STANJE V CPE (registre) IN V MEM (EX, MEM, WB)
   - ↳ UKAZA V IF in ID spremenita še PC za 8

2) SHRANI PC (naslov ukaza, ki je bil v ID stopnji) → NA VRH SKLADA → HW !!!

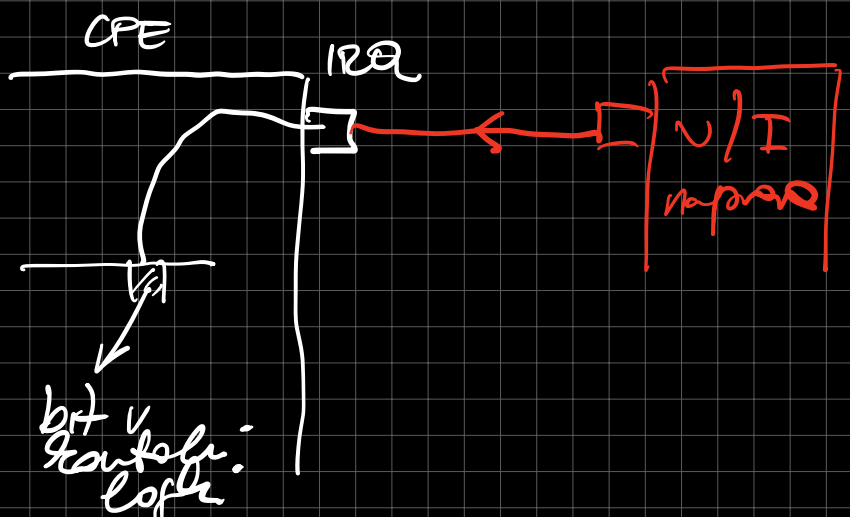
3) PC ← naslov PSP (npr. 0x00000000)
   - ↳ šedaj je novi PSP

KAKO SE VRNEMO iz PSP?

1. Preberi PC s sklada ukaz "POP PC"

Kako sprožimo prekinitveno obdelavo?

→ CPE ima priključek (PIN) za IRQ



Kaj pa, če je naprav več?

1. **Vse naprave si delijo IRQ**

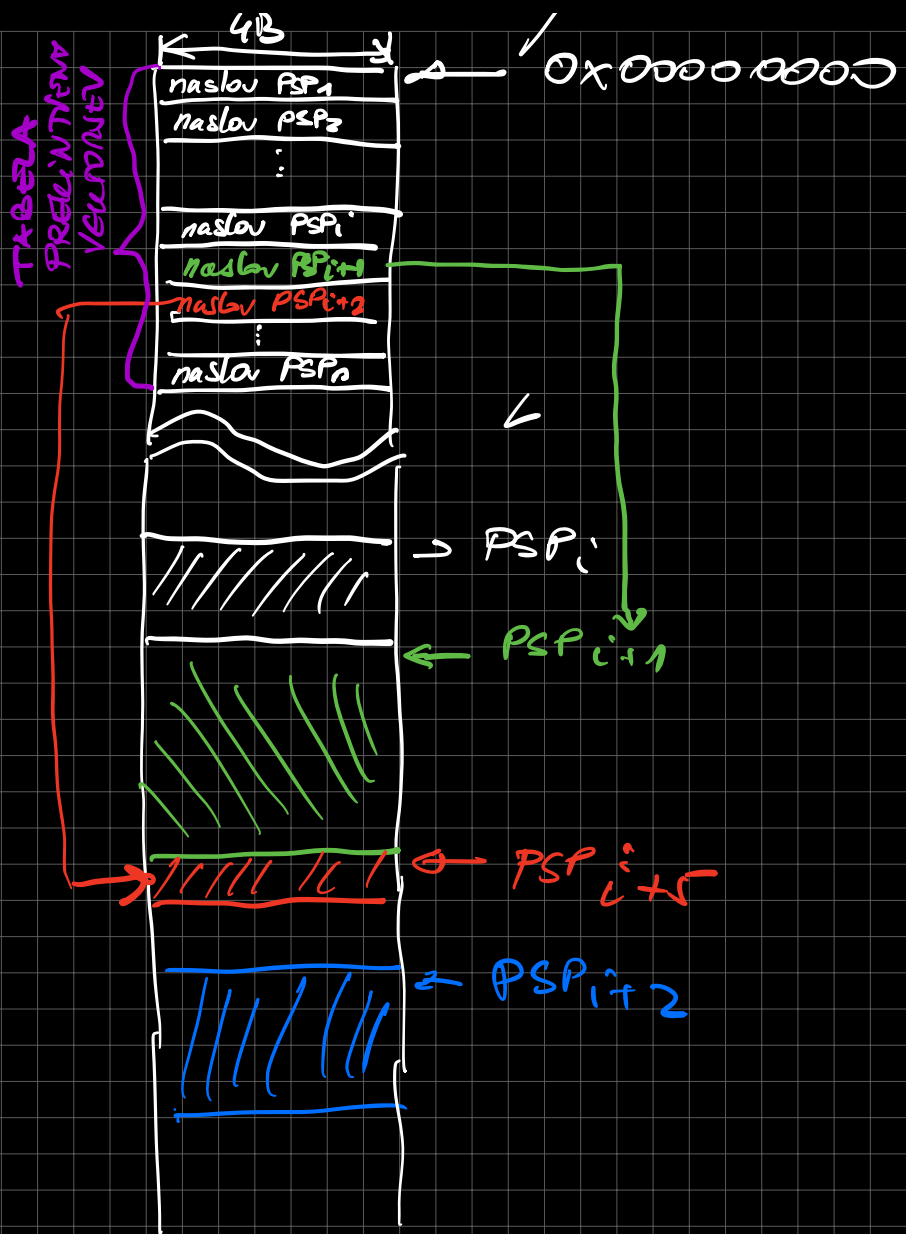
   ↳ CPE prebere (LOAD) statusne registre vseh V/I naprav

   ↳ PROGRAMSKO IZVRŠEVANJE PREKINITVENIH ROUTIN

   ~~PRIORITETA~~ ~~ZAHTEV~~

2. **Vsaka naprava ima svoj IRQ pin**

   → vsak IRQ vhod ima svoj PSP



če pride prek. zahtev na IRQi:

$PC \leftarrow M[4 \times IRQ_i]$

VEKTORSKE PREKINITVE

```
/*******************************************************************************
*
* Provide weak aliases for each Exception handler to the Default_Handler.
* As they are weak aliases, any function with the same name will override
* this definition.
*
*******************************************************************************/
   .weak      NMI_Handler
   .thumb_set NMI_Handler,Default_Handler

   .weak      HardFault_Handler
   .thumb_set HardFault_Handler,Default_Handler

   .weak      MemManage_Handler
   .thumb_set MemManage_Handler,Default_Handler

   .weak      BusFault_Handler
   .thumb_set BusFault_Handler,Default_Handler

   .weak      UsageFault_Handler
   .thumb_set UsageFault_Handler,Default_Handler

   .weak      SVC_Handler
   .thumb_set SVC_Handler,Default_Handler

   .weak      DebugMon_Handler
   .thumb_set DebugMon_Handler,Default_Handler

   .weak      PendSV_Handler
   .thumb_set PendSV_Handler,Default_Handler

   .weak      SysTick_Handler
   .thumb_set SysTick_Handler,Default_Handler

   .weak      WWDG_IRQHandler
   .thumb_set WWDG_IRQHandler,Default_Handler

   .weak      PVD_IRQHandler
   .thumb_set PVD_IRQHandler,Default_Handler

   .weak      TAMP_STAMP_IRQHandler
   .thumb_set TAMP_STAMP_IRQHandler,Default_Handler

   .weak      RTC_WKUP_IRQHandler
   .thumb_set RTC_WKUP_IRQHandler,Default_Handler
```

```
g_pfnVectors:
   .word  _estack
   .word  Reset_Handler
   .word  NMI_Handler
   .word  HardFault_Handler
   .word  MemManage_Handler
   .word  BusFault_Handler
   .word  UsageFault_Handler
   .word  0
   .word  0
   .word  0
   .word  0
   .word  SVC_Handler
   .word  DebugMon_Handler
   .word  0
   .word  PendSV_Handler
   .word  SysTick_Handler

   /* External Interrupts */
   .word     WWDG_IRQHandler                   /* Window WatchDog                  */
   .word     PVD_IRQHandler                    /* PVD through EXTI Line detection */
   .word     TAMP_STAMP_IRQHandler             /* Tamper and TimeStamps through the EXTI line */
   .word     RTC_WKUP_IRQHandler               /* RTC Wakeup through the EXTI line */
   .word     FLASH_IRQHandler                  /* FLASH                            */
   .word     RCC_IRQHandler                    /* RCC                              */
   .word     EXTI0_IRQHandler                  /* EXTI Line0                       */
   .word     EXTI1_IRQHandler                  /* EXTI Line1                       */
   .word     EXTI2_IRQHandler                  /* EXTI Line2                       */
   .word     EXTI3_IRQHandler                  /* EXTI Line3                       */
   .word     EXTI4_IRQHandler                  /* EXTI Line4                       */
   .word     DMA1_Stream0_IRQHandler           /* DMA1 Stream 0                    */
   .word     DMA1_Stream1_IRQHandler           /* DMA1 Stream 1                    */
   .word     DMA1_Stream2_IRQHandler           /* DMA1 Stream 2                    */
   .word     DMA1_Stream3_IRQHandler           /* DMA1 Stream 3                    */
   .word     DMA1_Stream4_IRQHandler           /* DMA1 Stream 4                    */
   .word     DMA1_Stream5_IRQHandler           /* DMA1 Stream 5                    */
   .word     DMA1_Stream6_IRQHandler           /* DMA1 Stream 6                    */
   .word     ADC_IRQHandler                    /* ADC1, ADC2 and ADC3s             */
   .word     CAN1_TX_IRQHandler                /* CAN1 TX                          */
   .word     CAN1_RX0_IRQHandler               /* CAN1 RX0                         */
   .word     CAN1_RX1_IRQHandler               /* CAN1 RX1                         */
   .word     CAN1_SCE_IRQHandler               /* CAN1 SCE                         */
   .word     EXTI9_5_IRQHandler                /* External Line[9:5]s              */
   .word     TIM1_BRK_TIM9_IRQHandler          /* TIM1 Break and TIM9              */
   .word     TIM1_UP_TIM10_IRQHandler          /* TIM1 Update and TIM10            */
   .word     TIM1_TRG_COM_TIM11_IRQHandler     /* TIM1 Trigger and Commutation and TIM11 */
   .word     TIM1_CC_IRQHandler                /* TIM1 Capture Compare             */
   .word     TIM2_IRQHandler                   /* TIM2                             */
   .word     TIM3_IRQHandler                   /* TIM3                             */
   .word     TIM4_IRQHandler                   /* TIM4                             */
   .word     I2C1_EV_IRQHandler                /* I2C1 Event                       */
```

Vector Number	Description	Type
0	Division by zero	Fault
1	Debug	Fault
2	NMI	Interrupt
3	Breakpoint	Trap
...	...	...
	...	...
14	Page Fault	Fault
...	...	...
32-255	External interrupts on INTR	Interrupt

```
Default_Handler:
Infinite_Loop:
    b  Infinite_Loop
```

```
IRQ_Handler:
        <handler instructions>
        ...
        ...
        subs pc, lr, #4      //  pc <- lr-4
```

```
        .org 0x00000000
Vector_Table:
        b Reset_Handler
        b Undefined_Handler
        b SWI_Handler
        b Prefetch_Handler
        b Abort_Handler
        nop                     // never used
        b IRQ_Handler
        b FIQ_HAndler


Reset_Handler:
        <handler instructions>
Undefined_Handler:
        <handler instructions>
SWI_Handler:
        <handler instructions>
Prefetch_Handler:
        <handler instructions>
Abort_Handler:
        <handler instructions>
IRQ_Handler:
        <handler instructions>
FIQ_Handler:
        <handler instructions>
```