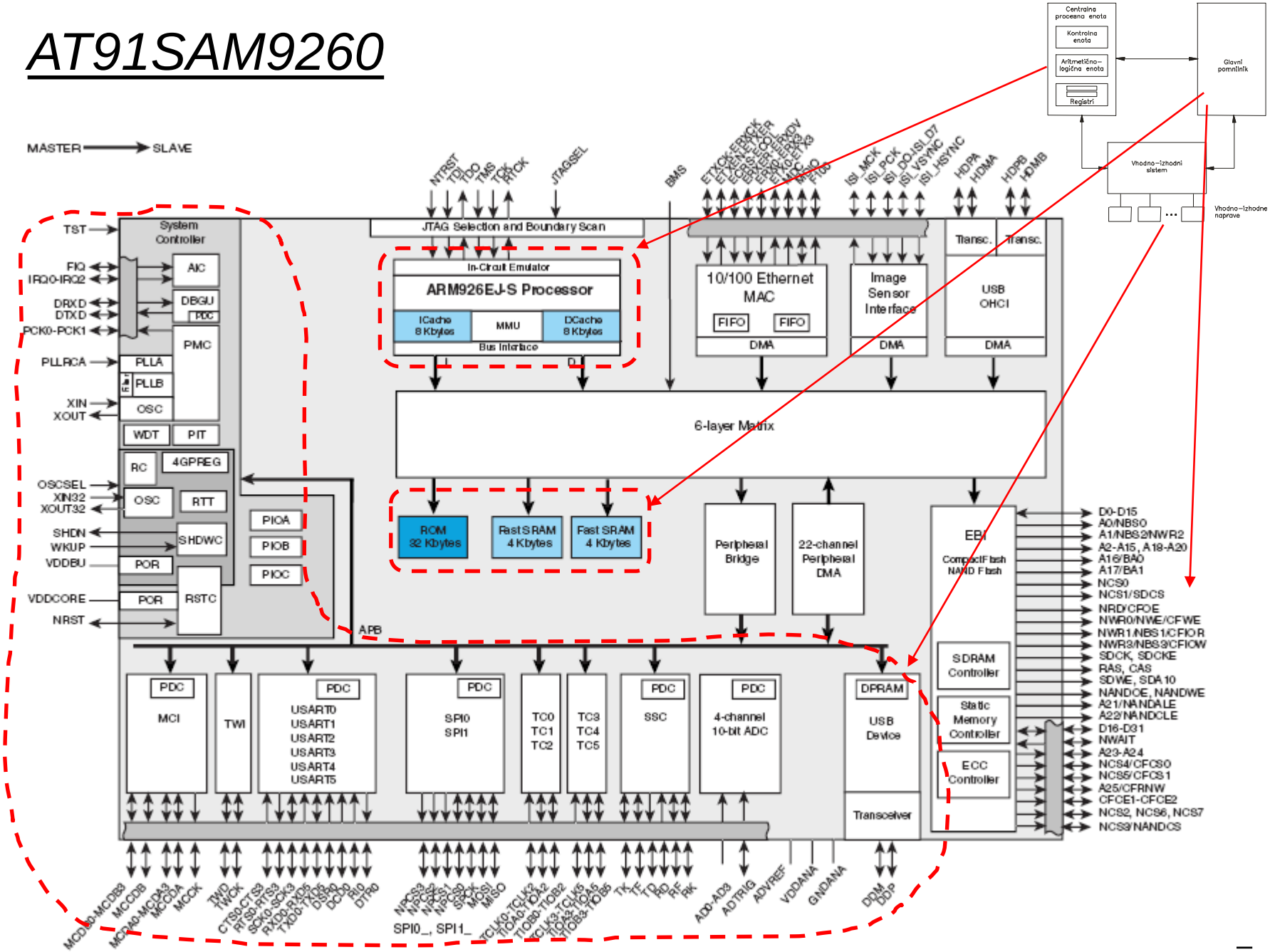
ARM

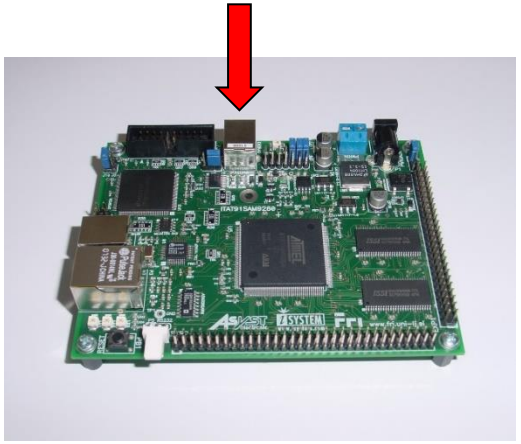
Vhodno / izhodne naprave

PIO Krmilnik

AT91SAM9260



Delo na FRI-SMS razvojnem sistemu



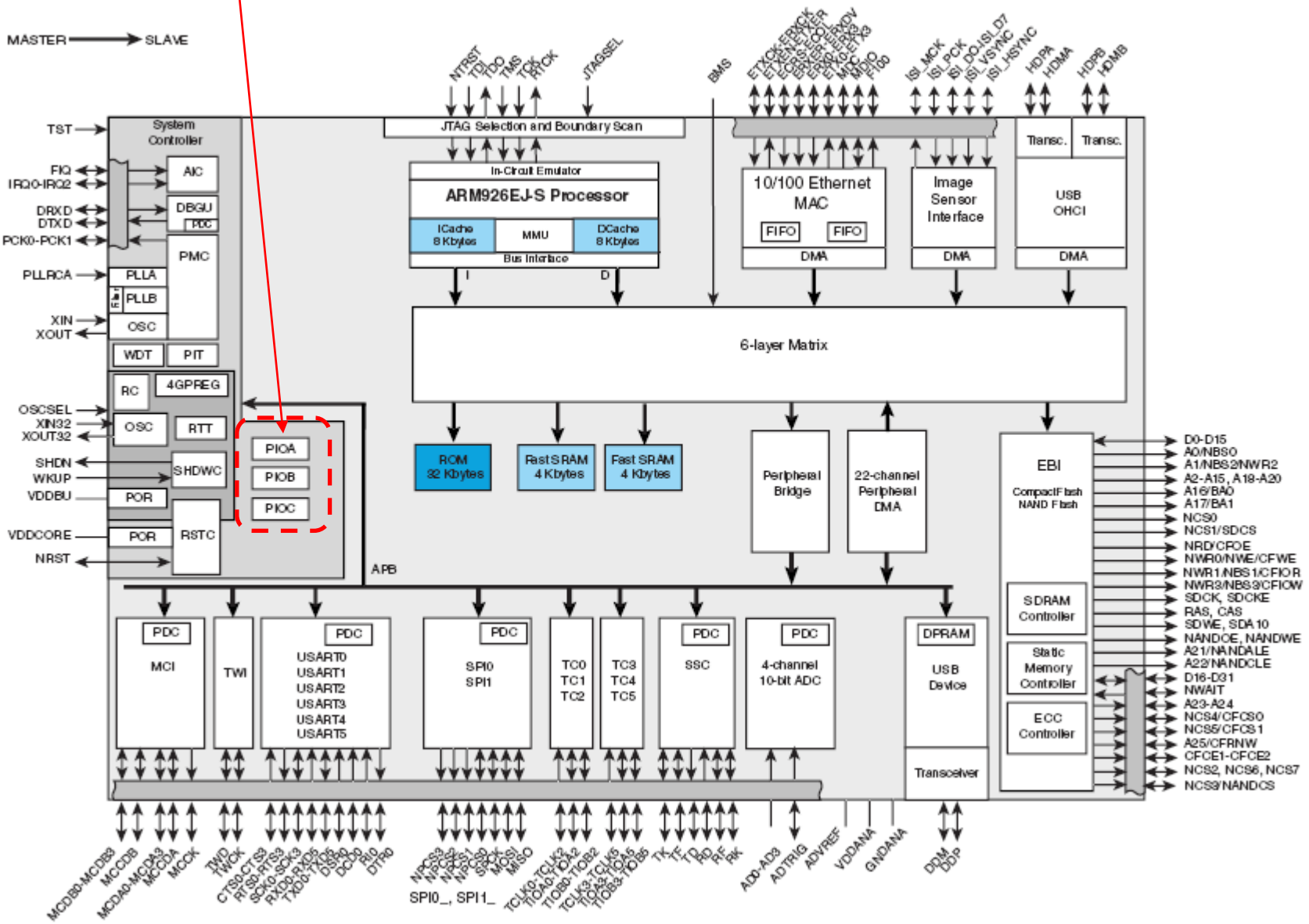
Priključitev :

- **USB** priklop na **daljši stranici**, sveti **zelena LED** dioda

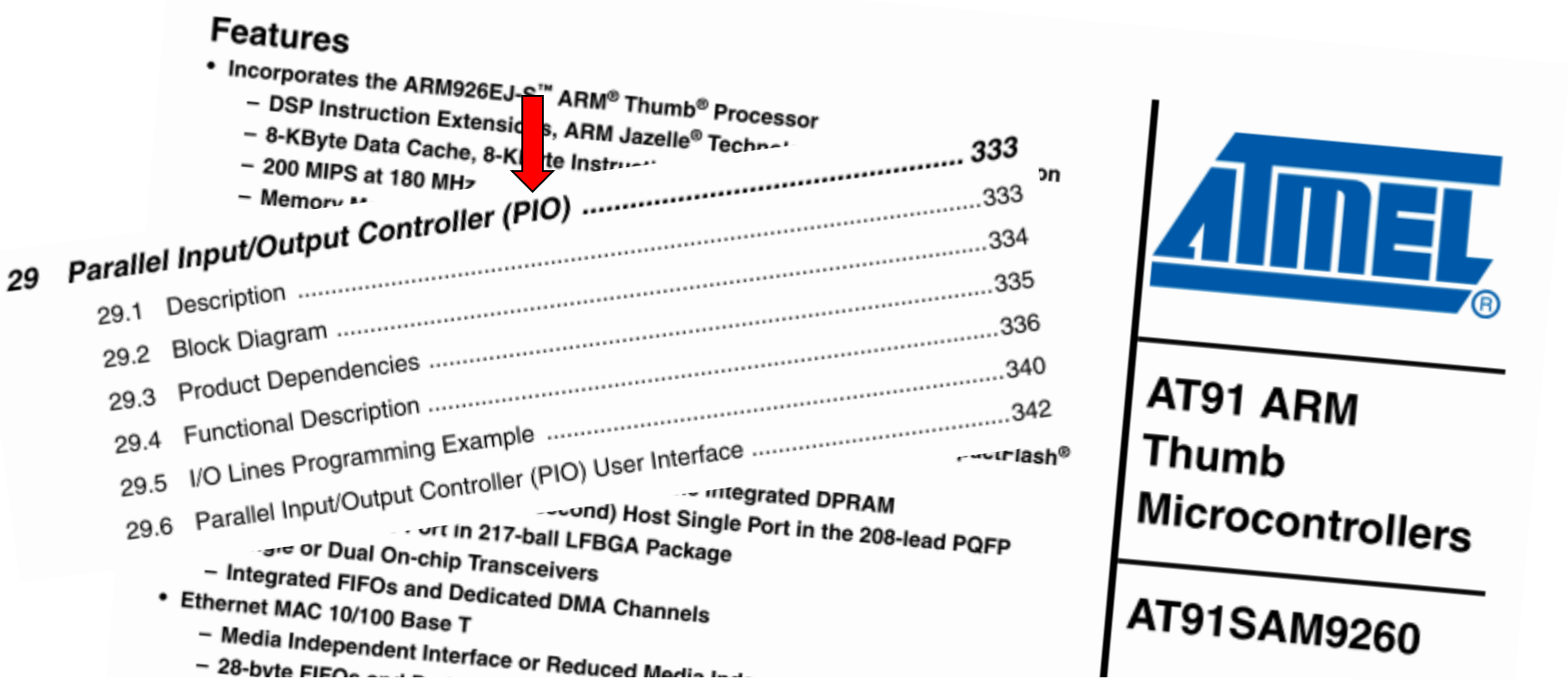
Poseben projekt za FRI-SMS (e-učilnica) :

- ***dodatne nastavitve*** (informativno) :
  - frekvenca urinega signala (višja poveča porabo!)
  - vklop predpomnilnikov
  - inicializacija sklada oz. SP – kazalca na sklad
- ***dodajanje vsebine (start.s):***
  - *podatki/operandi:*
    - dodamo v `/*constants*/` , končamo z .align
  - *program* :
    - dodamo v `/* enter your code here */`
    - na koncu programa je mrtva zanka
    - podprograme dodamo za mrtvo zanko

PIO Krmilnik



Vir: Podatkovna listina Atmel SAM 9260



Features

- Incorporates the ARM926EJ-S™ ARM® Thumb® Processor
  - DSP Instruction Extensions, ARM Jazelle® Technology
  - 8-KByte Data Cache, 8-KByte Instruction
  - 200 MIPS at 180 MHz
  - Memory ...

29 Parallel Input/Output Controller (PIO) 333

...Integrated DPRAM
...cond) Host Single Port in the 208-lead PQFP
...ort in 217-ball LFBGA Package
...gle or Dual On-chip Transceivers
  - Integrated FIFOs and Dedicated DMA Channels
- Ethernet MAC 10/100 Base T
  - Media Independent Interface or Reduced Media ...
  - 28-byte FIFOs ...

...on
...ash®

AT91 ARM Thumb Microcontrollers

AT91SAM9260

29. Parallel Input/Output Controller (PIO)

29.1 Description

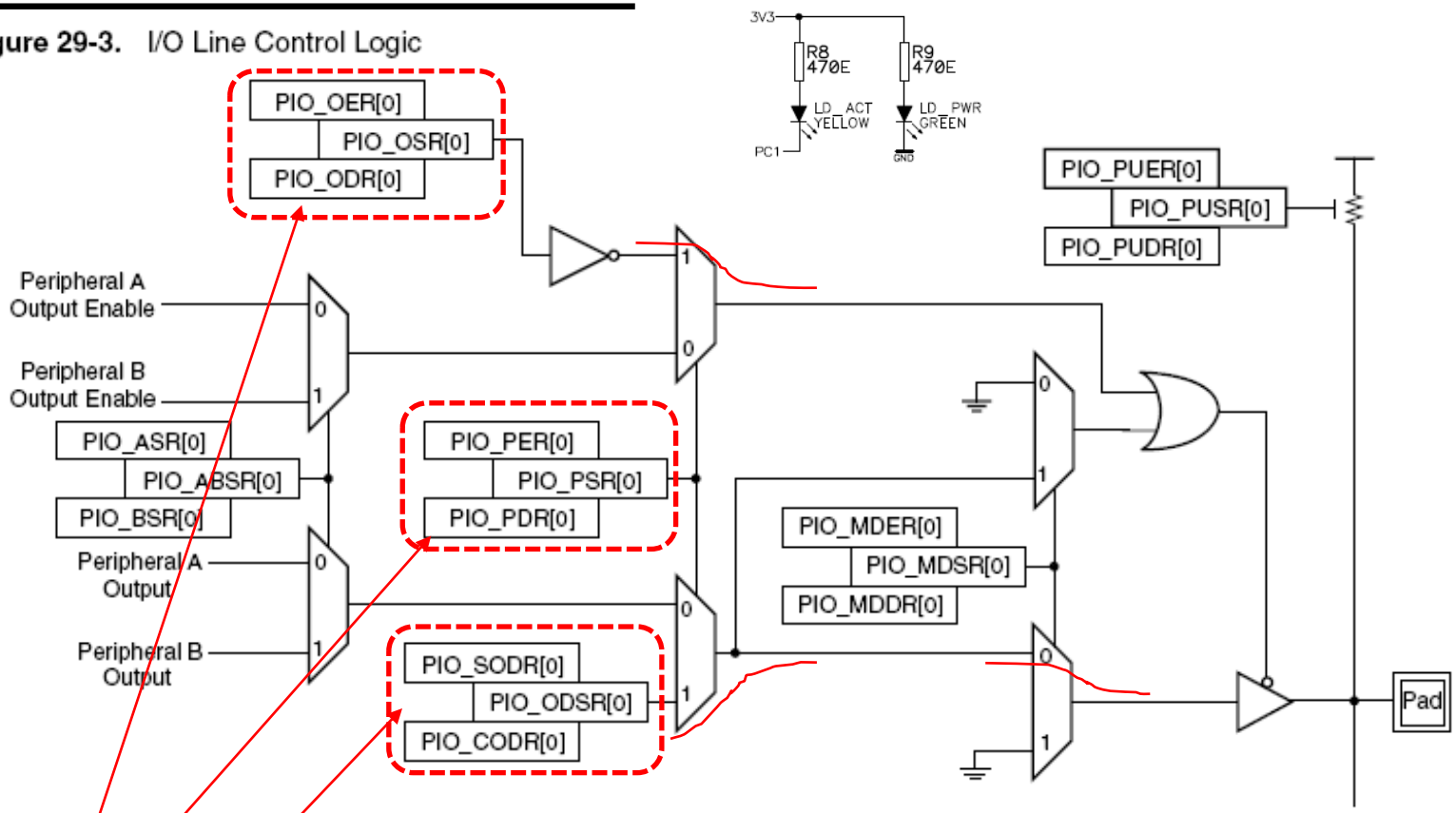
The Parallel Input/Output Controller (PIO) manages up to 32 fully programmable input/output lines. Each I/O line may be dedicated as a general-purpose I/O or be assigned to a function of an embedded peripheral. This assures effective optimization of the pins of a product.

Each I/O line is associated with a bit number in all of the 32-bit registers of the 32-bit wide User Interface.

Each I/O line of the PIO Controller features:

PIO krmilnik - izhod

Figure 29-3. I/O Line Control Logic



Registri z informacijo o stanju :

PSR (PIO Status Register): 1 - digitalni vhod/izhod / 0 - naprava AB

OSR (Output Status Register): 1 - izhod omogočen / 0 - izhod onemogočen (deluje kot vhod)

ODSR (Output Data Status Register): stanje izhoda 1 / 0

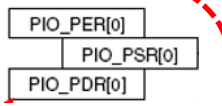
MDSR (Multiple Drive Status Register): tip izhoda 1 - 'open drain' / 0 - 'totem pole'

PUSR (Pull Up Status Register): 1 - omogočen / 0 - onemogočen; uporabno za vhode

PIO krmilnik - izhod

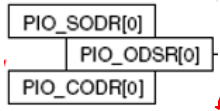
Registri za nastavitve delovanja:

PER (PIO Enable Register): 1 - nastavi kot digitalni vhod/izhod / 0 - ni spremembe delovanja
PDR (PIO Disable Register): 1 - nastavi kot napravo AB / 0 - ni spremembe delovanja



OER (Output Enable Register): 1 - nastavi kot izhod / 0 - ni spremembe delovanja
ODR (Output Disable Register): 1 - onemogoči izhod (dela kot vhod) / 0 - ni spremembe delovanja

SODR (Set Output Data Register): 1 - nastavi stanje izhoda na 1 / 0 - ni spremembe izhoda
CODR (Clear Output Data Register): 1 - nastavi stanje izhoda na 0 / 0 - ni spremembe izhoda



MDER (Multiple Drive Enable Register): 1 - nastavi izhod na 'totem pole' / 0 - ni spremembe delovanja
MDDR (Multiple Drive Disable Register): 1 - nastavi izhod na 'open drain' / 0 - ni spremembe delovanja

PUER (Pull Up Enable Register): 1 - omogočen 'pull-up' upor / 0 - ni spremembe delovanja
PUDR (Pull Up Disable Register): 1 - onemogočen 'pull-up' upor / 0 - ni spremembe delovanja

PIO krmilnik – krmiljenje izhodov

Spreminjanje bitov v V/I registrih:

Namesto branja-spreminjanja-pisanja enega registra imamo trojico registrov:

- en statusni (dejanski register) in
- dva pomožna s katerima postavljamo / brišemo bite:
  - ko v njima v določen bit vpišemo enico, s tem postavimo / brišemo ustrezen bit, ki krmili posamezno funkcijo,
  - stanje bita lahko preberemo v statusnem registru.

Potrebni koraki za krmiljenje izhoda:

1. vpiši 1 na ustrezno mesto v PER (določimo način delovanja kot vhod/izhod)
2. vpiši 1 na ustrezno mesto v OER (nastavimo kot digitalni izhod)
3. določi stanje izhoda s pisanjem v SODR / CODR (nastavljamo na 1/0)

Naslovi registrov:

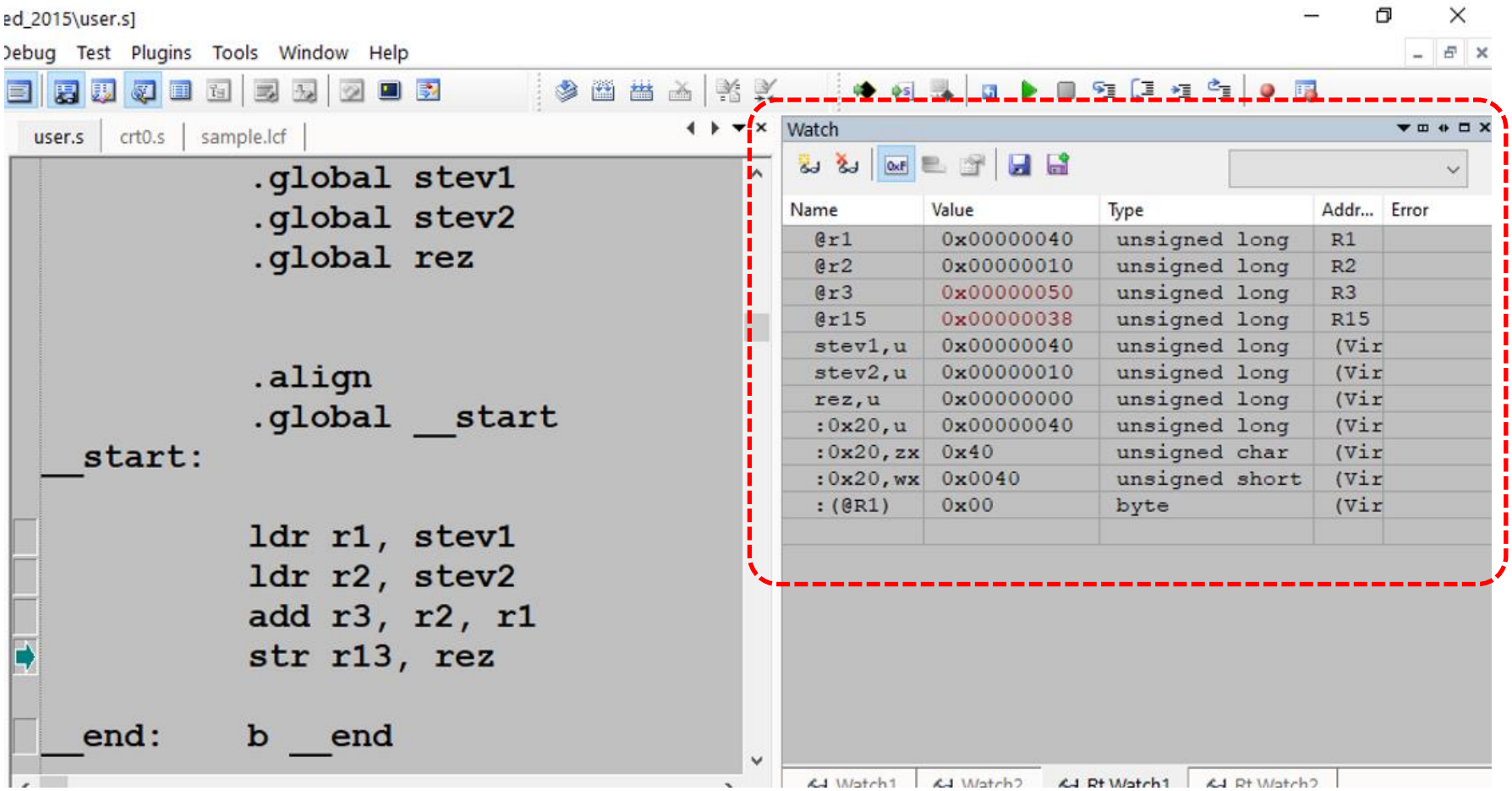
```
.equ PIOA_BASE, 0xFFFFF400 /* Začetek registrov za vrata A - PIOA */
.equ PIOB_BASE, 0xFFFFF600 /* Začetek registrov za vrata B - PIOB */
.equ PIOC_BASE, 0xFFFFF800 /* Začetek registrov za vrata C - PIOC */
.equ PIO_PER, 0x00       /* Odmiki... */
.equ PIO_OER, 0x10
.equ PIO_SODR, 0x30
.equ PIO_CODR, 0x34
```

PIO krmilnik – krmiljenje izhodov

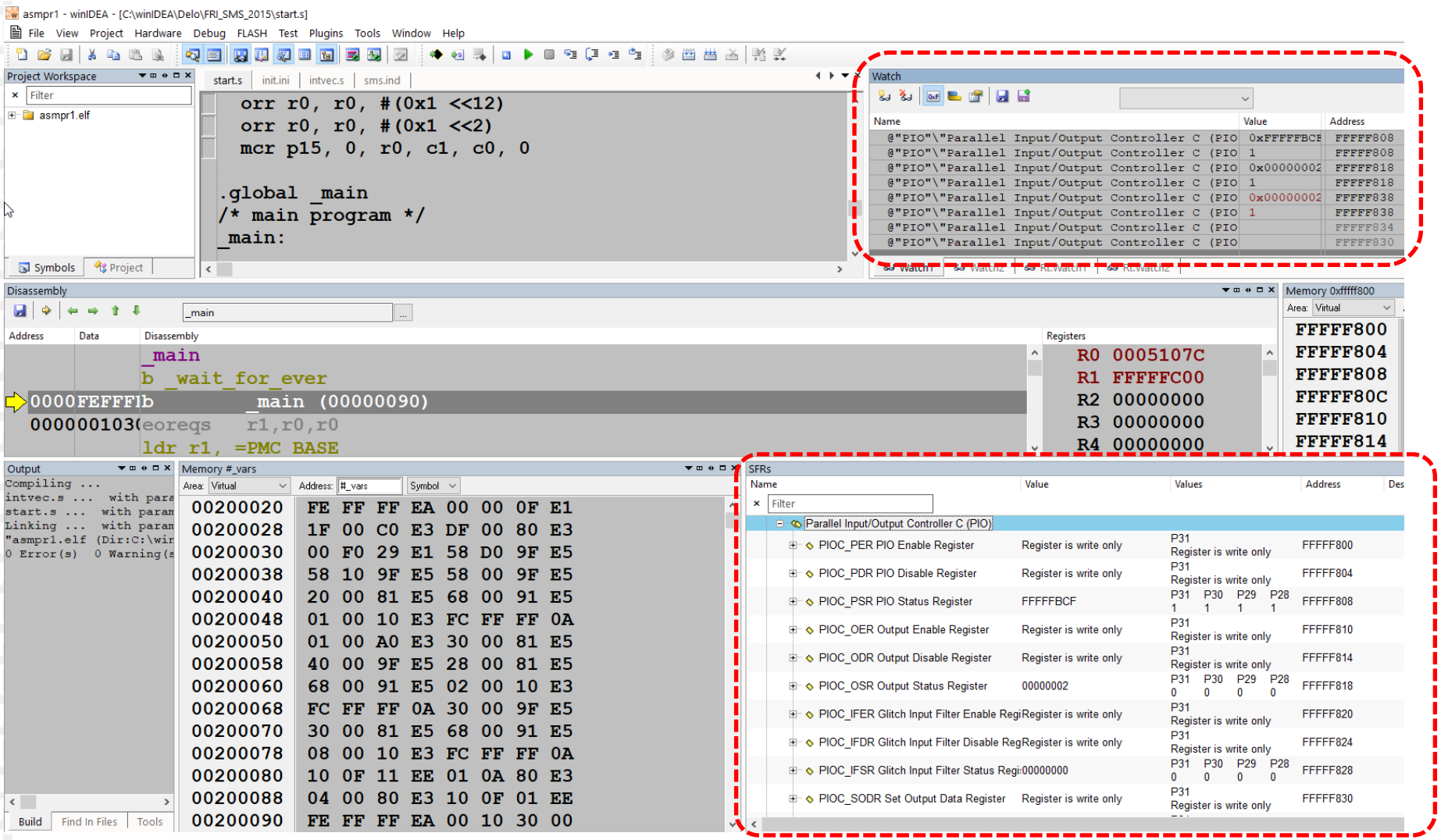
Zgled (za PB15):

```
.equ PIOB_BASE, 0xFFFFF600 /* Začetni naslov registrov za PIOB */
.equ PIO_PER, 0x00    /* Odmiki... */
.equ PIO_OER, 0x10
.equ PIO_SODR, 0x30
.equ PIO_CODR, 0x34

ldr r0, =PIOB_BASE
mov r1, #1 << 15
str r1, [r0, #PIO_PER]        /* Priključek B15 krmili PIO */
str r1, [r0, #PIO_OER]        /* Omogoči izhod na B15 */

str r1, [r0, #PIO_SODR]       /* Na priključek B15 zapiši stanje 1 */
str r1, [r0, #PIO_CODR]       /* Na priključek B15 zapiši stanje 0 */
```

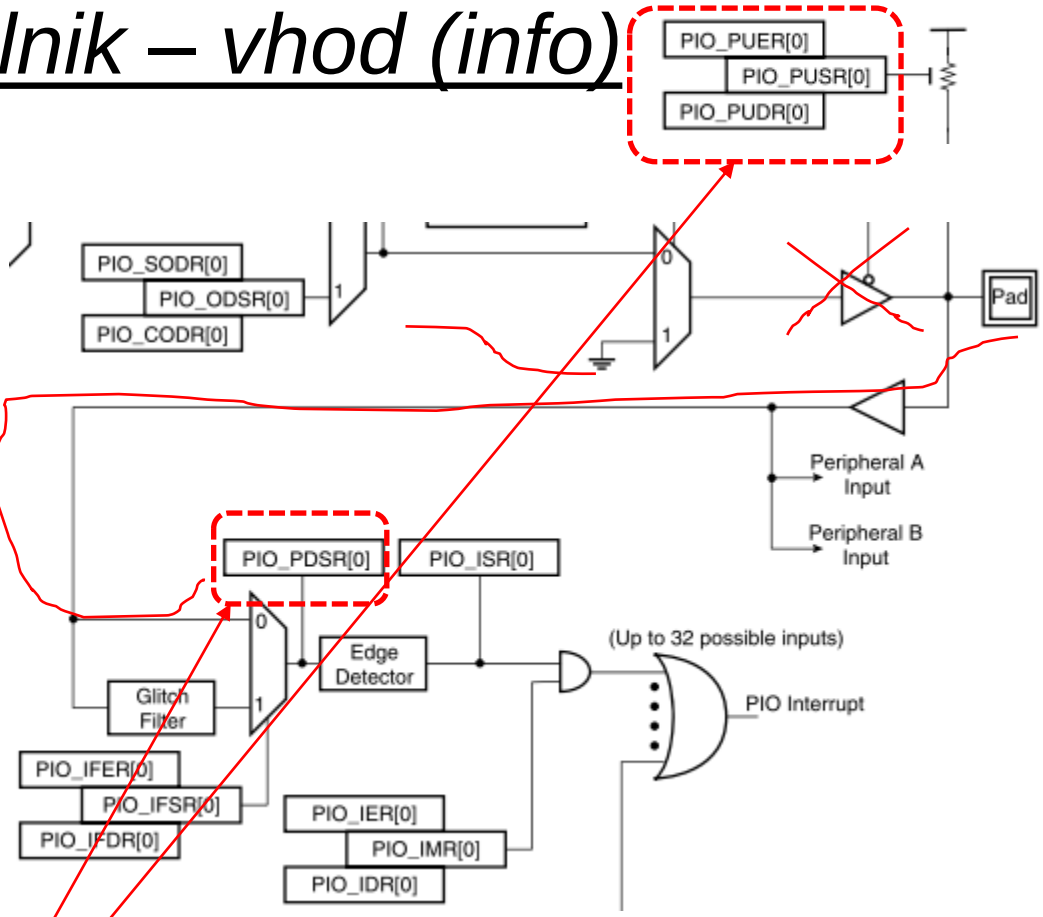
WinIdea – Watch okno



WinIdea – Watch okno – SFRs PIO krmilnik



PIO krmilnik – vhod (info)



Registri z informacijo o stanju :

PSR (PIO Status Register): 1 - digitalni vhod/izhod / 0 - naprava AB

OSR (Output Status Register): 1 - izhod omogočen / 0 - izhod onemogočen (deluje kot vhod)

ODSR (Output Data Status Register): stanje izhoda 1 / 0

MDSR (Multiple Drive Status Register): tip izhoda 1 - 'open drain' / 0 - 'totem pole'

PUSR (Pull Up Status Register): 1 - omogočen / 0 - onemogočen; uporabno za vhode

PDSR (Pin Data Status Register): stanje vhoda 1 / 0