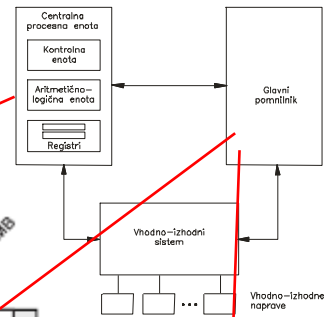
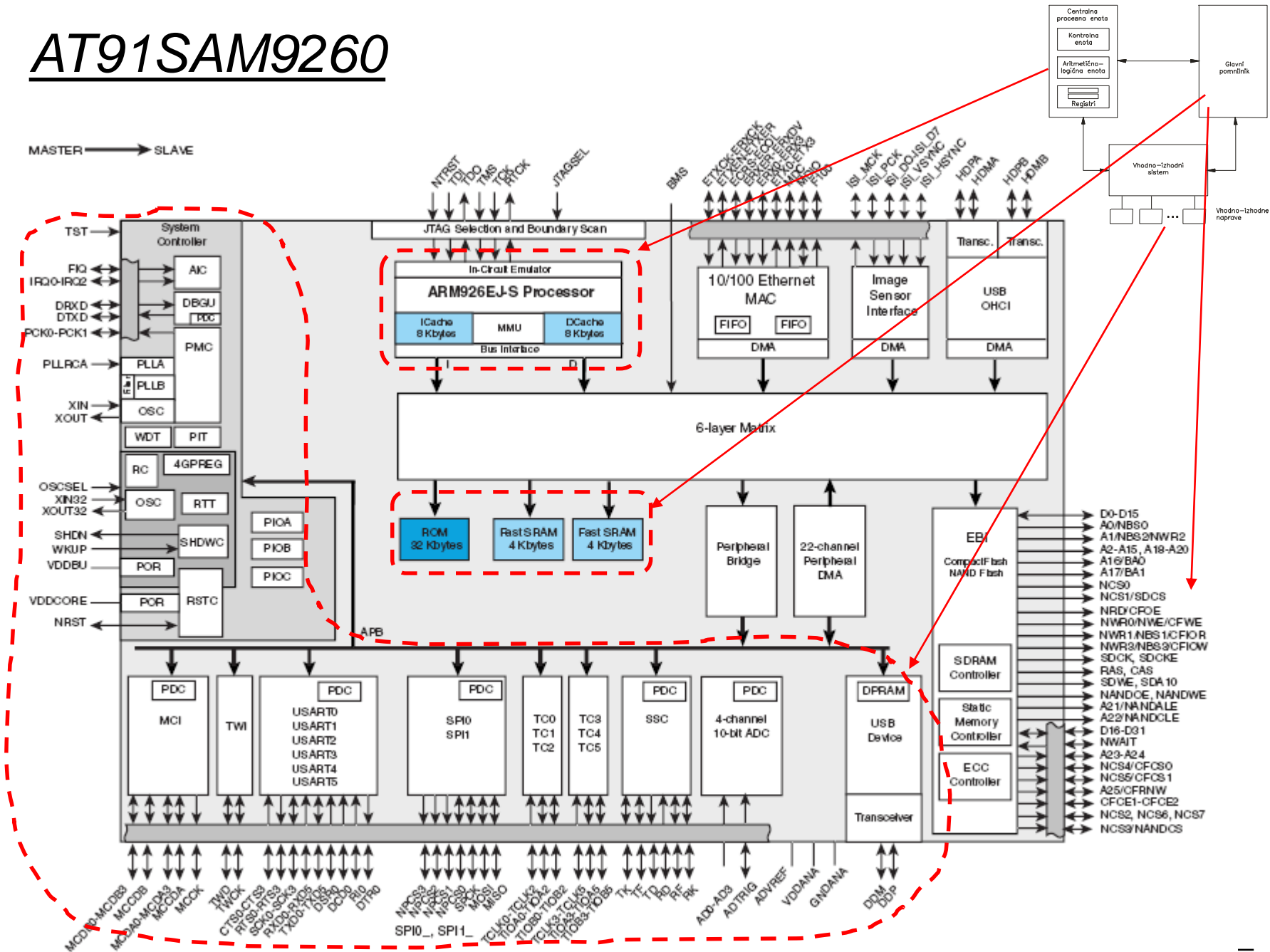


ARM

Vhodno / izhodne naprave

PIO Krmilnik

AT91SAM9260



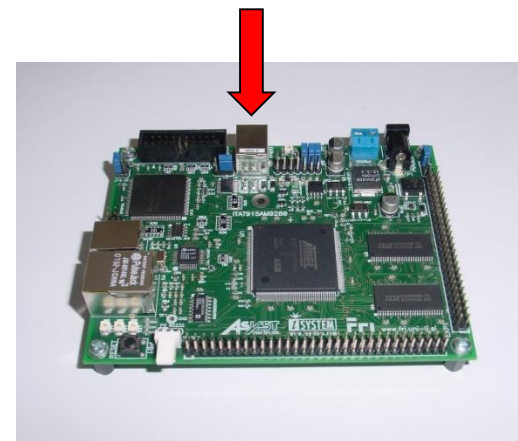
Delo na FRI-SMS razvojnem sistemu

Priključitev :

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

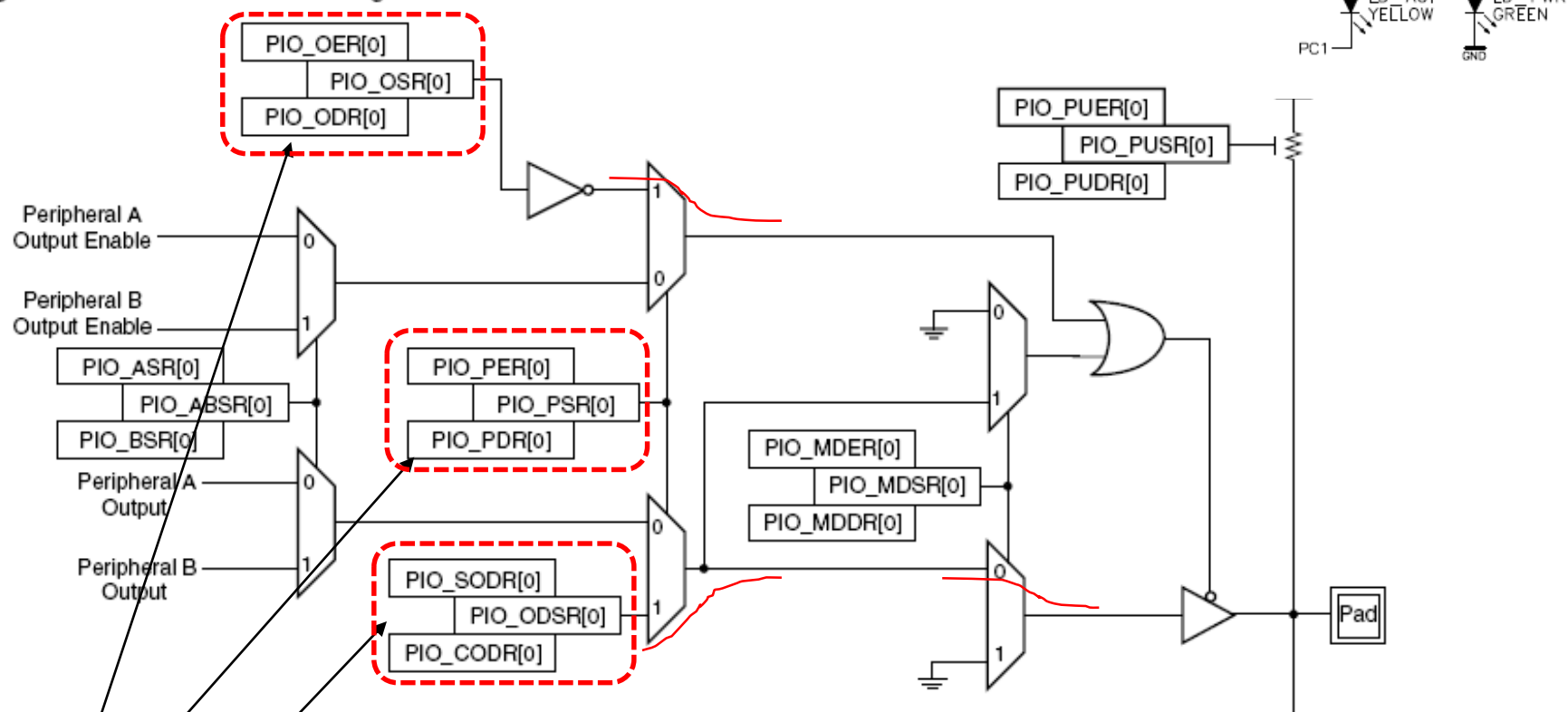
Poseben projekt za FRI-SMS (e-učilnica) :

- **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

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

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:

```
.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 */
```