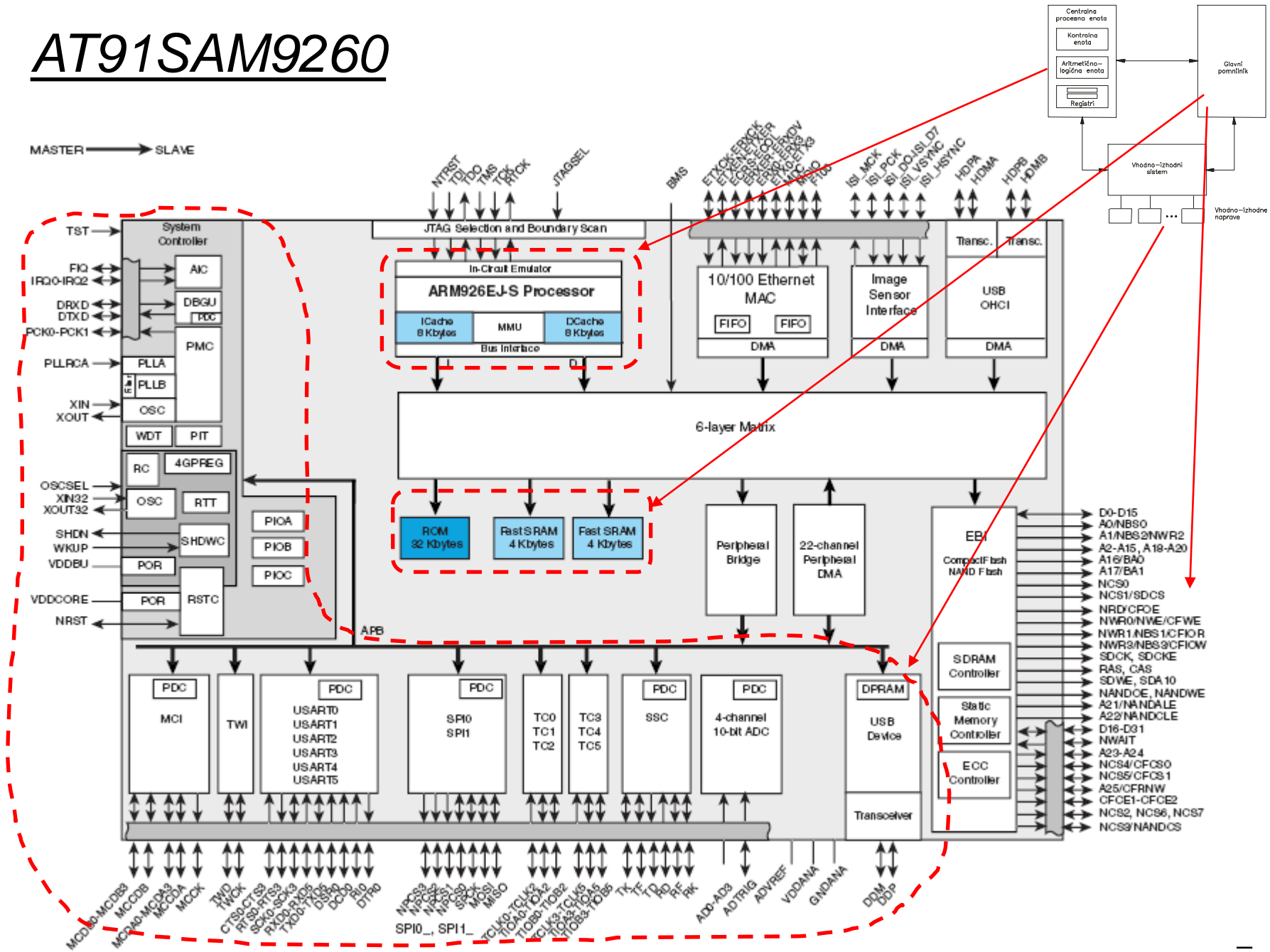


ARM

Vhodno / izhodne naprave

PIO Krmilnik

AT91SAM9260



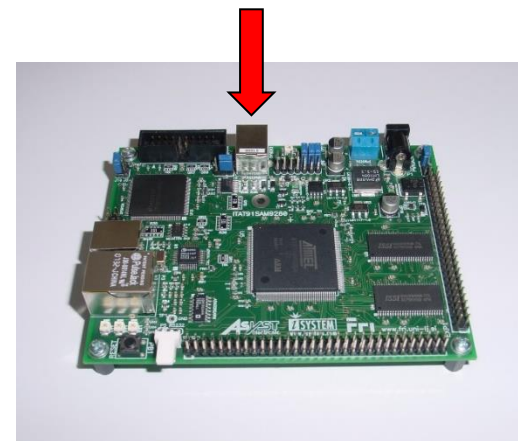
Delo na FRI-SMS razvojnem sistemu

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



Vir: Podatkovna listina Atmel SAM 9260

Features

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

29	Parallel Input/Output Controller (PIO)	333
29.1	Description	333
29.2	Block Diagram	334
29.3	Product Dependencies	335
29.4	Functional Description	336
29.5	I/O Lines Programming Example	340
29.6	Parallel Input/Output Controller (PIO) User Interface	342

- Ethernet MAC 10/100 Base T
 - Media Independent Interface or Reduced Media Int...
 - 28-byte FIFOs



29. Parallel Input/Output Controller (PIO)

29.1 Description

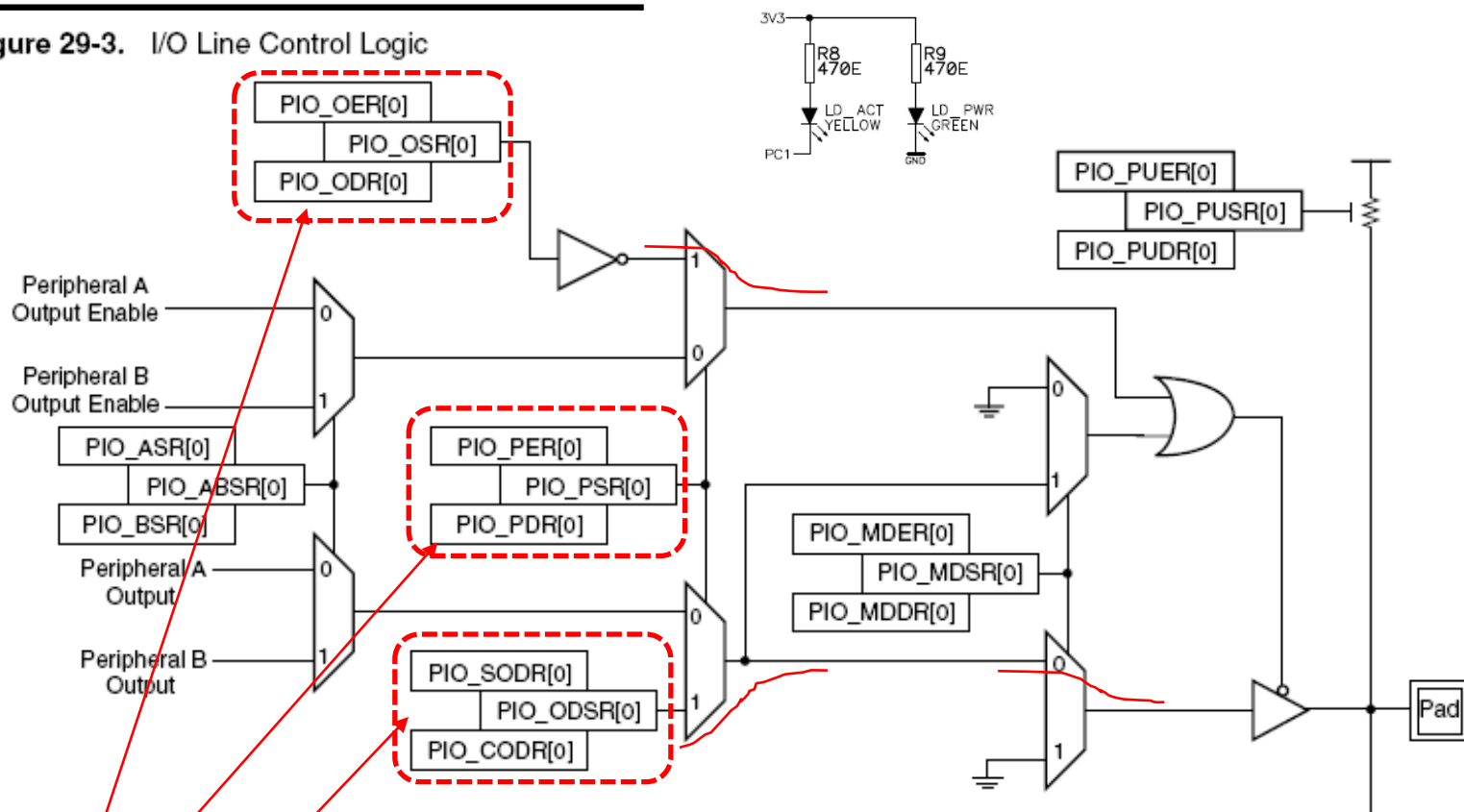
The Parallel Input/Output Controller (PIO) manages up to 32 fully programmable input/output lines. Each I/O line may be dedicated as a general-purpose I/O or be assigned to a function of an embedded peripheral. This assures effective optimization of the pins of a product.

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

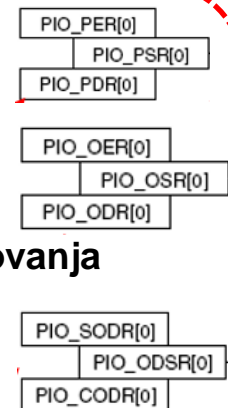
CODR (Clear Output Data Register): **1 - nastavi stanje izhoda na 0** / 0 - ni spremembe izhoda

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

ed_2015\user.s

Debug Test Plugins Tools Window Help

user.s crt0.s sample.lcf

```
.global stev1
.global stev2
.global rez

.align
.global __start

__start:

    ldr r1, stev1
    ldr r2, stev2
    add r3, r2, r1
    str r13, rez

end:    b __end
```

Watch

Name	Value	Type	Addr...	Error
@r1	0x00000040	unsigned long	R1	
@r2	0x00000010	unsigned long	R2	
@r3	0x00000050	unsigned long	R3	
@r15	0x00000038	unsigned long	R15	
stev1,u	0x00000040	unsigned long	(Vir	
stev2,u	0x00000010	unsigned long	(Vir	
rez,u	0x00000000	unsigned long	(Vir	
:0x20,u	0x00000040	unsigned long	(Vir	
:0x20,zx	0x40	unsigned char	(Vir	
:0x20 wx	0x0040	unsigned short	(Vir	
:(@R1)	0x00	byte	(Vir	

WinIDEA – Watch okno – SFRs PIO krmilnik

The screenshot displays the WinIDEA development environment with the following components:

- Project Workspace:** Shows the project 'asmpr1.elf'.
- Code Editor:** Contains assembly code:

```
orr r0, r0, #(0x1 <<12)
orr r0, r0, #(0x1 <<2)
mcr p15, 0, r0, c1, c0, 0

.global _main
/* main program */
_main:
```
- Disassembly:** Shows the disassembled code for the '_main' function:

```
Address      Data      Disassembly
main
b_wait_for_ever
0000FEFF1b   _main (00000090)
000000103e  eoreqs   r1,r0,r0
ldr r1, =PMC BASE
```
- Watch Window:** A table of variables being monitored:

Name	Value	Address
@"PIO"\Parallel Input/Output Controller C (PIO	0xFFFFFBCF	FFFFF808
@"PIO"\Parallel Input/Output Controller C (PIO	1	FFFFF808
@"PIO"\Parallel Input/Output Controller C (PIO	0x00000002	FFFFF818
@"PIO"\Parallel Input/Output Controller C (PIO	1	FFFFF818
@"PIO"\Parallel Input/Output Controller C (PIO	0x00000002	FFFFF838
@"PIO"\Parallel Input/Output Controller C (PIO	1	FFFFF838
@"PIO"\Parallel Input/Output Controller C (PIO		FFFFF834
@"PIO"\Parallel Input/Output Controller C (PIO		FFFFF830
- Registers:** A list of CPU registers:

Register	Value
R0	0005107C
R1	FFFFFFC0
R2	00000000
R3	00000000
R4	00000000
- Memory #_vars:** A table of memory addresses and their contents:

Address	#_vars	Symbol
00200020	FE FF FF EA 00 00 0F E1	
00200028	1F 00 C0 E3 DF 00 80 E3	
00200030	00 F0 29 E1 58 D0 9F E5	
00200038	58 10 9F E5 58 00 9F E5	
00200040	20 00 81 E5 68 00 91 E5	
00200048	01 00 10 E3 FC FF FF 0A	
00200050	01 00 A0 E3 30 00 81 E5	
00200058	40 00 9F E5 28 00 81 E5	
00200060	68 00 91 E5 02 00 10 E3	
00200068	FC FF FF 0A 30 00 9F E5	
00200070	30 00 81 E5 68 00 91 E5	
00200078	08 00 10 E3 FC FF FF 0A	
00200080	10 0F 11 EE 01 0A 80 E3	
00200088	04 00 80 E3 10 0F 01 EE	
00200090	FE FF FF EA 00 10 30 00	
- SFRs PIO Controller:** A detailed view of the Parallel Input/Output Controller registers:

Name	Value	Values	Address	Des
PIOC_PER PIO Enable Register	Register is write only	P31 Register is write only	FFFFF800	
PIOC_PDR PIO Disable Register	Register is write only	P31 Register is write only	FFFFF804	
PIOC_PSR PIO Status Register	FFFFFFBCF	P31 P30 P29 P28 1 1 1 1	FFFFF808	
PIOC_OER Output Enable Register	Register is write only	P31 Register is write only	FFFFF810	
PIOC_ODR Output Disable Register	Register is write only	P31 Register is write only	FFFFF814	
PIOC_OSR Output Status Register	00000002	P31 P30 P29 P28 0 0 0 0	FFFFF818	
PIOC_IFER Glitch Input Filter Enable Register	Register is write only	P31 Register is write only	FFFFF820	
PIOC_IFDR Glitch Input Filter Disable Register	Register is write only	P31 Register is write only	FFFFF824	
PIOC_IFSR Glitch Input Filter Status Register	00000000	P31 P30 P29 P28 0 0 0 0	FFFFF828	
PIOC_SODR Set Output Data Register	Register is write only	P31 Register is write only	FFFFF830	

