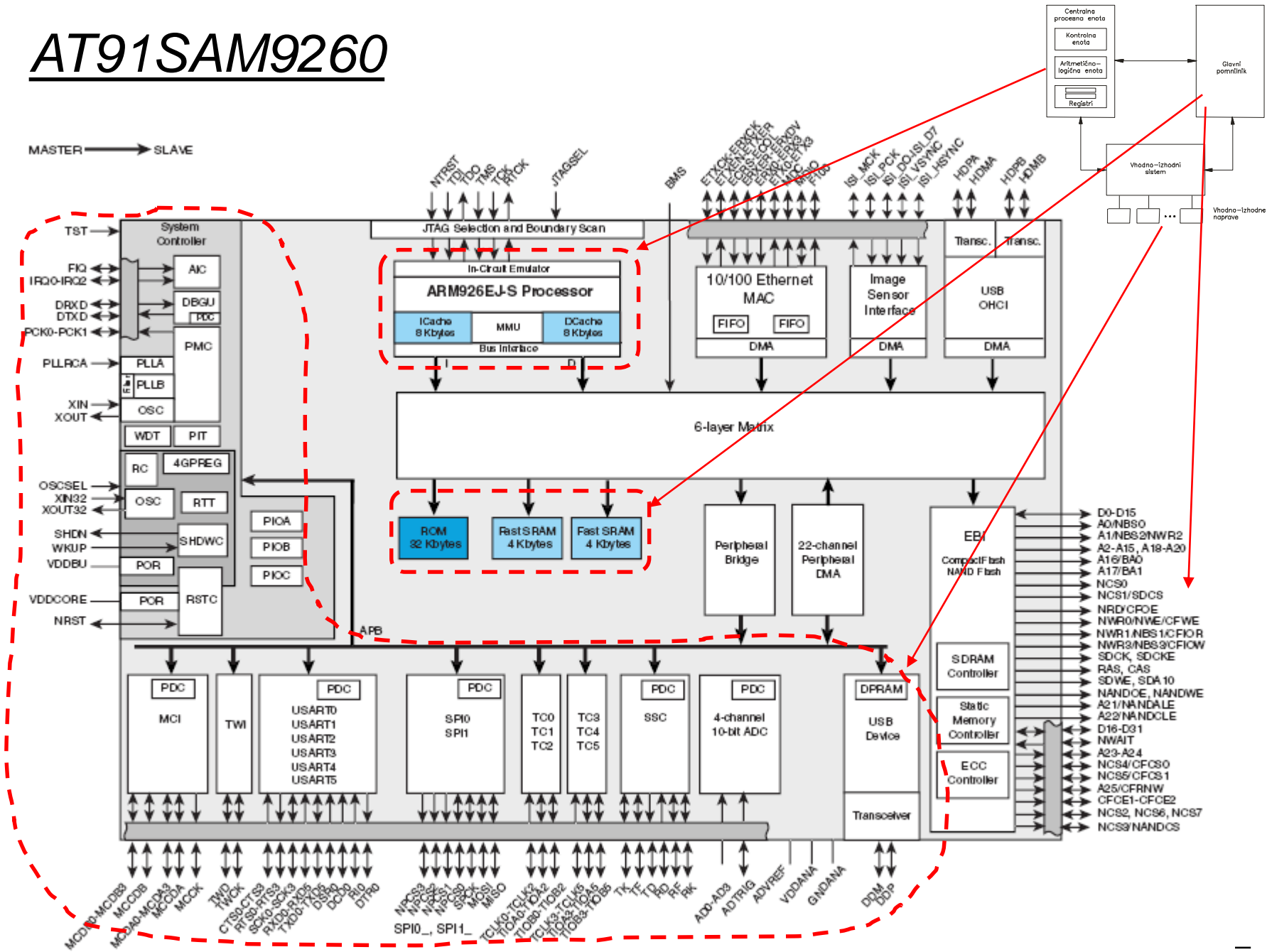


ARM

Projekt za FRI-SMS vgrajen sistem

winIDEA

AT91SAM9260



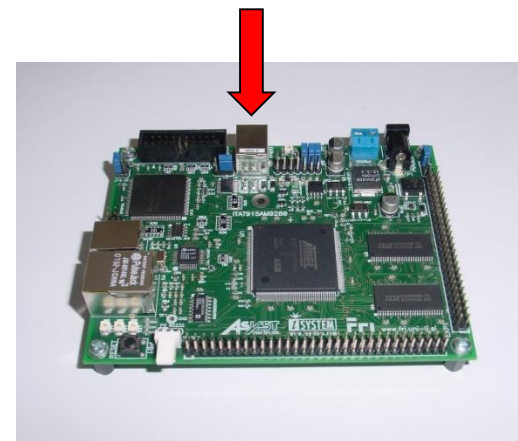
Delo na FRI-SMS razvojnem sistemu

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



Inicializacija sistema – začetno stanje

intvec.s :

```
.text
.code 32

.global _start
B _start          /* RESET INTERRUPT */
B _error          /* UNDEFINED INSTRUCTION INTERRUPT */
B _error          /* SOFTWARE INTERRUPT */
B _error          /* ABORT (PREFETCH) INTERRUPT */
B _error          /* ABORT (DATA) INTERRUPT */
B _error          /* RESERVED */
B _error          /* IRQ INTERRUPT */
B _error          /* FIQ INTERRUPT */

.end
```

Način delovanja, sklad

start.s :

_start:

/ select system mode*

```
    CPSR[4:0]      Mode
  [ 11111      System  ]
*/
```

`mrs r0, cpsr`

`bic r0, r0, #0x1F` */* clear mode flags */*

`orr r0, r0, #0xDF` */* set supervisor mode (0b11111) + DISABLE IRQ, FIQ
I=F=1, T(humb)=0 */*

`msr cpsr, r0`

/ init stack */*

`ldr sp, _lstack_end`

Sistemska ura

start.s :

```
/* setup system clocks */
ldr r1, =PMC_BASE

ldr r0, = 0x0F01
str r0, [r1,#CKGR_MOR]      /* Main Osc. enable, 15*8 cycles for osc. setup */

osc_lp:
ldr r0, [r1,#PMC_SR]      /* Main oscillator is stabilized ? */
tst r0, #0x01
beq osc_lp

mov r0, #0x01
str r0, [r1,#PMC_MCKR]    /* CSS: Master Clock Selection - Main Clock*/

ldr r0, =0x2000bf00 | ( 124 << 16) | 12 /* 18,432 MHz * 125 / 12 = 192 MHz*/
str r0, [r1,#CKGR_PLLAR] /* Clock Generator PLL A Register */

pll_lp:
ldr r0, [r1,#PMC_SR]      /* PLL A is locked ? */
tst r0, #0x02
beq pll_lp
```

Sistemska ura II + predpomnilniki

start.s :

```
/* MCK = PCK/4 */  
ldr r0, =0x0202  
str r0, [r1,#PMC_MCKR] /* CSS = PLLA, MCK=PCK/4 */
```

mck_lp:

```
ldr r0, [r1,#PMC_SR] /* MCKRDY: Master Clock Status ? */  
tst r0, #0x08  
beq mck_lp
```

```
/* Enable caches */
```

```
mrc p15, 0, r0, c1, c0, 0 /* Move to ARM register from coprocessor */  
orr r0, r0, #(0x1 <<12)  
orr r0, r0, #(0x1 <<2)  
mcr p15, 0, r0, c1, c0, 0 /* Move to coprocessor from ARM registers*/
```

Glavni program,



podprogrami,



spremenljivke.



start.s :

```
.global _main
/* main program */
_main:

/* user code here */
    bl INIT_IO
LOOP:
    bl LED_ON
    ldr r0,=500
    bl DELAY
    bl LED_OFF
    ldr r0,=500
    bl DELAY

    b LOOP
/* end user code */
_wait_for_ever: b _wait_for_ever
/* subroutines here */
/* constants */

_1stack_end: .long __STACK_END__
```