

```
mov r1, r4, LSL #2 ; r1=r4*4
ali r1, r4, LSR #2 ; r1=r4/2^2
```

```
PSP:
sub r14,r14,#4
stmfd r13!, {r0-r3,r14}
```

```
ldr r0,=NIZ
ldr r1,=NIZ1
ldr r3,=N
ldr r2,[r3]
bl CHANGE
ldr r0,=NIZ1
ldr r3,=N
ldr r2,[r3]
bl SND_DMA
ldr r0,=NIZ
ldr r3,=N
ldr r2,[r3]
RCV_DMA
ldr r0,=AIC_BASE
str r0, [r0, #AIC_EOICR]
ldmfd r13!, {r0-r3,pc}^
```

```
INIT_DBGU:
stmfd sp!,{r0-r1,lr}
ldr r0,=DBGU_BASE
mov r1, #1 << 11
/* normalen način delovanja, */
str r1,[r0,#DBGU_MR]
/* brez parnosti */
mov r1,#26
str r1,[r0,#DBGU_BRGR]
/* hitrost na 115200 baud */
mov r1,#0b01010000
str r1,[r0,#DBGU_CR]
/* omogoči TX in RX */
mov r1, #1
str r1, [r0, #DBGU_IER]
```

```
/* 26=MCK/(16*BAUD_RATE) ...
MCK=48Mhz=48*10^6,
BAUD_RATE=115200 */
```

### Potrebni koraki za nastavitve UART:

1. Nastavi način delovanja (normal mode) in parnost z vpisom v DBGU\_MR
2. Nastavi hitrost prenosa z vpisom v DBGU\_BRGR vrednost, ki jo vpišemo izračunamo po formuli  $MCK/(16*BAUD\_RATE)$
3. Omogoči oddajnik in sprejemnik z vpisom 1 v bita TXEN in RXEN v DBGU\_CR

### Sprejem / oddaja znakov preko UART:

- ob sprejemu znaka se na 1 postavi bit RXRDY v DBGU\_SR
- znak preberemo v spodnjih 8 bitih DBGU\_RHR
- znak, ki ga želimo poslati, vpišemo v spodnjih 8 bitov DBGU\_THR in sicer takrat, ko je oddajnik pripravljen na oddajo (bit TXRDY v DBGU\_SR postavljen na 1)

### Potrebni koraki za zaznavanje prehodov števca 0xFFFF -> 0x0000:

1. Števca omogoči urin signal z vpisom na ustrezno mesto v PMC\_PCER
2. Izberi frekvenco urinega signala z vpisom v spodnje 3 bite TCx\_CMR
3. Omogoči uro z vpisom 1 v CLKEN (bit 0 v TCx\_CCR)
4. Sproži števec z vpisom 1 v SWTRG (bit 2 v TCx\_CCR)
5. Čakaj na zastavico COVFS (bit 0 v TCx\_SR)

### Potrebni koraki za štetje do vrednosti v RC:

1. Števca omogoči urin signal z vpisom na ustrezno mesto v PMC\_PCER
2. Izberi frekvenco urinega signala z vpisom v spodnje 3 bite TCx\_CMR
3. Z vpisom WAVE in WAVSEL (1 in npr. 10) nastavi delovanje števca\*
4. Zapiši ustrezno mejo v RC\*
5. Omogoči uro z vpisom 1 v CLKEN (bit 0 v TCx\_CCR)
6. Sproži števec z vpisom 1 v SWTRG (bit 2 v TCx\_CCR)
7. Čakaj na zastavico CPCS\*

### 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)

```
INIT_LED:
stmfd sp!,{r0-r1,lr}
ldr r0,=PIOC_BASE
mov r1, #1 << 1
str r1, [r0, #PIO_PER]
/* Priključek C1 krmili PIO */
str r1, [r0, #PIO_OER]
/* Omogoči izhod na C1 */
ldmfd sp!,{r0-r1,lr}
LED_ON:
stmfd sp!,{r0-r1,lr}
ldr r0,=PIOC_BASE
mov r1, #1 << 1
str r1, [r0, #PIO_CODR]
/* Na priključek C1 zapiši stanje 0 */
ldmfd sp!,{r0-r1,pc}
```

```
INIT_TC0_PSP:
stmfd r13!,{r0-r1,r14}
ldr r0,=PMC_BASE
mov r1,#1<<17
str r1,[r0,#PMC_PCER]
ldr r0,=TC0_BASE
ldr r1,=0x7004
str r1,[r0,#TC_CMR]
ldr r1,=10000
str r1,[r0,#TC_RC]
mov r1,#1
str r1,[r0,#TC_CCR]
mov r1,#1<<2
str r1,[r0,#TC_CCR]
mov r1, #16
str r1,[r0,#TC_IER]
ldmfd r13!,{r0-*/
```

```
ldr r0, [r1, r2, LSL #2]
; r0<-mem32[r1+r2*4]
```

```
INIT_AIC:
stmfd r13!, {r0-r2,r14}
ldr r0,=AIC_BASE
mov r1, #1 << 17
mov r2, #4
```

```
/* high level sensitive, PRI
str r2, [r0, #AIC_SMR17]
ldr r2,=ENABLE_IRQ
str r2, [r0, #AIC_SVR17]
/* set interrupt vector for IS17 */
str r1, [r0, #AIC_IER]
/* enable interrupt for IS17 */
str r0, [r0, #AIC_EOICR]
/* slepo pisanje v EOICR */
ldmfd r13!, {r0-r2,pc}INIT_TC0_PSP:
stmfd r13!,{r0-r1,r14}
```

```
RCV_DMA:
stmfd sp!,{r0-r2,lr}
adr r0, N
ldr r1, [N]
ldr r0,=polje
```

```
cmp r1, #1
movlt r1, #70
cmp r1, #80
movgt r1, #70
```

```
ldr r2,=DBGU_BASE
str r0, [r2, DMA_RPR]
str r1, [r1, DMA_RCR]
mov r1,#1
str r1,[r0,#DMA_PTCR]
ldmfd sp!,{r0-r2,pc}
```

```
SNR_DMA:
stmfd sp!,{r0-r2,lr}
ldr r2,=DBGU_BASE
str r0, [r2, DMA_TPR]
str r1, [r2, DMA_TCR]
mov r1, #1<<8
str r1, [r2, DMA_PTCR]
ldmfd sp!,{r0-r2,pc}
```

```
ENABLE_IRQ:
stmfd r13!, {r0, r14}
mrs r0, cpsr
and r0,r0,#127
/* Omogoci prekinitve */
msr cpsr_c,r0
ldmfd r13!, {r0, pc}
```

3. Imamo SDRAM s 4 moduli organizacije 2048x256x32,  $t_{RD} = 18ns$ ,  $t_{RAS} = 42ns$ ,  $t_{RP} = 18ns$  in  $CL = 3ns$ . Frekvenca ure je  $f_{CLK} = 100 MHz$  in uporabljamo izvorno sinhronski prenos.

- a) Določite časovno zaporedje ukazov za branje 4 zaporednih besed iz ene vrstice SDRAM-a. Upoštevajte, da je potrebno vrstico po končanem branju zapreti.
- b) Izračunajte minimalen čas prenosa potreben za branje cele vrstice v modulu. Zakasnitev na prenosnih poteh zanemarite.

a) Potrebno zaporedje je naslednje:

- odpiranje vrstice (izbira vrstice in banke)
- zakasnitev  $t_{RD}$  zaokrožena na najmanjše celo število urinih period, ki je enako ali večje  $t_{RD}$  čipa (v našem primeru 2 u.p. ali 20ns >= 18ns)
- branje vrstice (izbira stolpca)
- zakasnitev  $t_{CL}$  (v našem primeru 3 u.p. ali 30ns)
- prenos 4 besed po vodilu (4 u.p. ali 40ns)
- zapiranje vrstice
- pred ponovnim odpiranjem iste vrstice mora preteči še zakasnitev  $t_{BP}$  (v našem primeru 2 u.p. ali 20ns >= 18ns)

b) Čas prenosa (v u.p.) =  $t_{RD}$  (v u.p.) +  $t_{CL}$  (v u.p.) + N (u.p.) = 2u.p.+3u.p.+256u.p. = 261 u.p. = 2,61μs kjer je N velikost vrstice v besedah

### a) asinhronski prenos

Vsak prenos 32-bitov traja 2 urini periodi

$$t_0 = 1/100 MHz = 10 ns$$

$$B = 4B/2t_0 = 4B / (2*10ns) = 200 MB/s$$

### b) eksplozijski prenos

Tu za vsako vrstico potrebujemo 4 urine periode

+ 1 začetna = 5

$$B = 16B/5t_0 = 16B / (5*10ns) = 320 MB/s$$

### c) protokolski eksplozijski prenos

Za prenos vseh vrstic potrebujemo 16\*16/4

urinih period + 1 začetna = 65

$$B = 256B/65t_0 = 256B / (65*10ns) = 393,85 MB/s$$

393,85/200 = 1,97 protokolski eksplozijski

prenos ie za 97% hitrejši od asinhronskega

$$a) BW_{ef} = f * N * \frac{L^{4U}}{240 + 3 + 5} = 66 * 10^6 * 4B * 0,968 = 255,48MB / s$$

Efektivna pasovna širina je teoretična ( $f^*N$ ) pasovna širina, kjer upoštevamo, da pri vsakem prenosu zgubimo 3 urine periode v naslovni fazi in v povprečju 5 urinih period pri čakanju na vodilo.

b) V enem eksplozijskem prenosu lahko naprava dostavi/prebere le en blok podatkov (512B).

$$t_{prenosa} = \frac{\left( \frac{\text{velikost\_bloka}}{\text{sirina\_vodila}} + \text{naslovna\_faza} + \text{cakanje\_na\_vodilo} \right) * \text{dolzina\_prenosa}}{\text{velikost\_bloka}} * \text{f\_vodila}$$

$$t_{prenosa} = \frac{\left( \frac{512B}{4B} + 3 + 5 \right) * 512 * 10^3 B}{66 * 10^6 Hz} = 2,06ms$$

Prenosna hitrost = 512000B / 0.00206s = 248,47 MB/s