Univerza v Ljubljani
Fakulteta za računalništvo
in informatiko

ORGANIZACIJA RAČUNALNIKOV

2. DOMAČA NALOGA – ODDAJNIK MORSEJEVE KODE

Luka Šveigl, 63200301

OPIS PROJEKTA:

V sklopu 2. domače naloge pri predmetu Organizacija računalnikov smo morali napisati aplikacijo, ki uporablja tehnologije FRI-SMS ali STMi, prav tako pa uporablja različne nivoje programiranja, C ali zbirnik. Sam sem izbral eno izmed ponujenih tem, to je oddajnik morsejeve kode, ki sem jo realiziral na razvojnem sistemu FRI-SMS z zbirnikom.

Osnovna zahteva za projekt je bila uporaba vhodno/izhodnih naprav, ki smo jih spoznali pri predmetu. Pri svojem projektu sem uporabil napravo časovnik TC0, katerega sem uporabljal za zakasnitev vžiga in izklopa LED diode, in napravo DMA krmilnik za oddajanje ter sprejemanje niza znakov, ki se je pretvoril v morsejevo abecedo.

Sam projekt deluje tako, da najprej inicializira vhodno/izhodne naprave, nato pa v načinu local loopback sprejema niz znakov, ki je definiran v kodi. Le ta niz po sprejemu z uporabo definiranih podprogramov, ki bodo opisani v naslednjih poglavjih, spremeni v morsejevo abecedo in ga s pomočjo LED diode tudi prikaže. Local loopback način sem izbral, ker sem ta projekt delal doma, in nisem imel pri sebi potrebnega kabla RS232. Na vajah načrtujem projekt preoblikovati, da bo delal preko terminala, za kar je potrebna le zamenjava klicov RCV_DMA in SND_DMA.

Za vklop in izklop LED diode sem uporabljal podprograma LED_ON in LED_OFF, ki smo jih definirali na vajah, za zakasnitev sem uporabljal podprogram DELAY_TC0, ki smo ga prav tako definirali na vajah. Za pošiljanje ter sprejemanje znakov pa sem uporabljal podprograma RCV_DMA in SND_DMA, katera smo tudi napisali na vajah.

1. NALOGA:

Napišite podprogram XMCHAR, ki dobi parameter v registru r0. Parameter je lahko '.' ali '-'. V primeru, da je parameter '.' naj podprogram za približno 150 ms prižge LED (LED_ON, DELAY). V primeru, da je parameter '-' naj podprogram za približno 300 ms prižge LED (LED_ON, DELAY). V obeh primerih naj nato ugasne LED in počaka približno 150ms (LED_OFF, DELAY). Podprogram preizkusite s primernim glavnim programom.

```
XMCHAR:
  stmfd sp!, {r0, lr}

  cmp r0, #'.'    @ Check if R0 is .
  moveq r0, #150  @ If R0 is ., delay for 150 ms

  cmp r0, #'-'    @ Check if R0 is -
  moveq r0, #300  @ If R0 is -, delay for 300 ms

  bl LED_ON       @ Turn LED on and wait
  bl DELAY_TC0

  mov r0, #150    @ Turn LED off and wait for 150 ms
  bl LED_OFF
  bl DELAY_TC0

  ldmfd sp!, {r0, pc}
```

Program XMCHAR deluje tako, da ohranja vrednost registra R0. Program najprej preveri, ali je vrednost v R0 enaka “.” ali “-” in nato vanj premakne vrednost 150 ali 300, ki predstavlja število milisekund zakasnitve. Program nato pokliče program LED_ON, da prižge LED diodo, nato pa z časovnikom TC0 počaka prej določeno število sekund. V R0 se nato shrani vrednost 150, LED dioda se ugasne z klicom podprograma LED_OFF, nato pa z uporabo časovnika TC0 zopet program zakasnimo.

2. NALOGA:

Napišite podprogram XMCODE, ki kot parameter v r0 dobi kazalec na z ničlo zaključen niz, v katerem se pojavljajo samo '.' in '-'. Konec niza označuje 0. Podprogram naj gre skozi niz znak po znak in za vsak znak pokliče podprogram iz XMCHAR. Ko naleti na 0, pa naj počaka približno 300 ms in se zaključi. Podprogram preizkusite tako, da oddate zaporedje "--.-".

```
XMCODE:
  stmfd sp!, {r0, r1, lr}
  mov r1, r0      @ Copy address of string from R0 to R1

  CLOOP:          @ Loop through string
    ldrb r2, [r1] @ Get char of string
    cmp r2, #0    @ Check if end of string - if true, quit subroutine
    beq CBYE

    mov r0, r2
    bl XMCHAR     @ Call XMCHAR to turn on LED

    add r1, r1, #1@ Move to next char in string
    b CLOOP

  CBYE:
  mov r0, #300    @ When done, wait 300 ms
  bl DELAY_TC0
  ldmfd sp!, {r0, r1, pc}
```

Program XMCODE deluje tako, da ohranja vrednost registrov R0 in R1. Program najprej kopira vrednost R0 v R1, nato pa se v zanki sprehodi čez niz, na katerega kaže naslov v R1. V vsaki iteraciji preveri, ali je trenutni znak 0 (konec niza), in če ni pokliče program XMCHAR, da prižge LED diodo na podlagi vrednosti v R0. Po koncu oddajanja znaka program počaka 300 milisekund.

3. NALOGA:

Napišite podprogram GETMCODE, ki kot parameter v r0 dobi ASCII kodo velike tiskane črke 'A' – 'Z'. Podprogram naj v r0 vrne kazalec na z ničlo zaključen niz, v katerem je zaporedje črk in pik, ki ustreza Morsejevi kodi ustrezne črke. Podprogram naj uporablja tabelo, v kateri so Morsejeve kode črk in je vsaka črka predstavljena s šestimi znaki. Tiste kode, ki so krajše, naj imajo na koncu več 0. Npr. koda črke 'A' bo oblike:

.ascii ".-" @ A

.byte 0,0,0,0

.ascii "-…" @ B

.byte 0,0

```
GETMCODE:
  stmfd sp!, {r1, lr} @ Store only value of R1 and link register

  mov r2, #6      @ Move value to multiply R1 with
                  @ Value must be 6, as morse codes
                  @ are stored at address intervals of 6

  sub r0, r0, #65 @ Reduce value of char by 65 - get index
  mul r1, r0, r2  @ Multiply value by 6 - get address deviation

  adr r0, CHARS   @ Get address of morse codes
  add r0, r0, r1  @ Move to correct offset

  ldmfd sp!, {r1, pc} @ Load only value of R1 and program counter
```

Program GETMCODE deluje tako, da ohranja vrednost samo registra R1. Program najprej v register R2 premakne vrednost 6, ki jo kasneje uporabi za množenje. Program nato od vrednosti v registru R0 (črka, torej vrednost med 65 in 90), odšteje 65, da dobi indeks črke, katerega potrebuje za iskanje morsejevega zapisa. Program nato to novo vrednost pomnoži s 6, da dobi odmik od začetnega naslova morsejevih kod, shranjenih na naslovu CHARS. Nato se v register R0 shrani naslov CHARS. Program nato naslovu v R0 prišteje prej izračunan odmik trenutnega znaka.

4. NALOGA:

Napišite podprogram XWORD, ki kot parameter v r0 dobi kazalec na z ničlo zaključen niz, v katerem so samo velike črke. Podprogram naj s klicanjem GETMCODE in XMCODE po Morsejevi kodi pošlje črko za črko, ko pride do zaključne 0 pa počaka približno 1 sekundo in se zaključi. Za preizkus oddajte "SOS".

```
XWORD:
  stmfd sp!, {lr} @ Store only value of link register
  mov r1, r0      @ Copy address from R0 to R1

  XLOOP:          @ Loop through all characters in word
    ldrb r0, [r1] @ Load character
    cmp r0, #0    @ Check if end of word
    beq XBYE      @ If end of word, quit subroutine
    bl GETMCODE   @ Get pointer to start of morse code
    bl XMCODE     @ Display morse code
    add r1, r1, #1@ Move to next character in string
    b XLOOP

  XBYE:
  ldr r0, =1000   @ Wait for 1 s
  bl DELAY_TC0
  ldmfd sp!, {pc} @ Load only value of program counter
```

Program XWORD deluje tako, da ne ohranja vrednosti nobenih registrov. Program najprej premakne naslov niza, shranjen v registru R0 v register R1, nato pa se v zanki sprehodi čez celoten niz, dokler ne sreča končne ničle. Za vsak znak, ki ni 0, se pokliče program GETMCODE, ki črko pretvori v naslov njene morsejeve komponente, nato pa se pokliče program XMCODE, ki ta naslov uporabi za prikaz te črke preko LED diode. Ko program prikaže celotno besedo, se sproži zakasnitev 1 sekunde.

5. NALOGA:

Napišite glavni program, v katerem preko zaporednega vmesnika enote DBGU sprejmete besedo (znake sprejemajte, dokler ne sprejmete znaka za tipko enter). Nato to besedo v Morsejevi kodi 'oddajte' preko LED diode.

```
bl INIT_IO  @ Init input/output devices
bl INIT_TC0 @ Init timer 0
bl DEBUG_INIT

INF_LOOP:                  @ Local loopback testing loop
  adr r0, STRING           @ Get address of string
  add r0, r0, #0x200000
  mov r1, #7               @ Set number of characters
  bl RCV_DMA               @ Receive data

  ldr r3, =DBGU_BASE
  CHECK_ENDRX:             @ Check if receiving finished
    ldr r2, [r3, #DBGU_SR]
    tst r2, #0b10000
    beq CHECK_ENDRX

   adr r0, STRING          @ Get address of string
   add r0, r0, #0x200000
   mov r1, #7              @ Set number of characters
   bl SND_DMA              @ Send data

   ldr r3, =DBGU_BASE
   CHECK_ENDTX:            @ Check if sending finished
    ldr r2, [r3, #DBGU_SR]
    tst r2, #0b10000
    beq CHECK_ENDTX

   bl XWORD                @ Display morse code on LED
   b INF_LOOP
```

Glavni program, kot je prikazan na zgornji sliki, najprej inicializira potrebne vhodno/izhodne naprave, nato pa v zanki sproži delovanje Local loopback načina. Kot že prej omenjeno, sem ta način uporabil, ker sem ta projekt delal doma, na vajah pa bom kodo popravil, da bo delovala preko terminala. Ko so podatki sprejeti in poslani, se pokliče program XWORD, ki prikaže morsejevo kodo niza, sprejetega z krmilnikom DMA. Ta zanka se ponavlja v neskončnost.