Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

University of Zagreb

Faculty of Electrical Engineering and Computing

Arhitektura računala 1

*Computer Architecture 1*

**Laboratorijska vježba broj 2**

*Laboratory exercise no. 2*

Marina Rupe

0036483027

# Zadatak / Exercise

**4. Kućni alarm: šifra preko tipaka**

U novosagrađenu kuću vlasnik je stavio alarmni sustav zasnovan na FRISC3 procesoru. Dva reed-prekidača (Velleman HAA27) postavljeni su na ulazna vrata i na vrata terase.

Alarm se aktivira unošenjem kombinacije od 4 unaprijed definirana znaka (koji se unose pritiskom na tipke: lijevo, desno,gore ili dolje) i pritiskom na srednju tipku za potvrdu, nakon čega vlasnik ima 10 s da zatvori oba vrata i napusti kuću.

Kad vlasnik uđe u kuću odnosno kad se bilo koja od vrata otvore, alarm mora davati kratke zvučne signale tražeći unos šifre. Šifra se unosi preko tipkala i sastoji se od kombinacije 4 unaprijed definirana znaka i pritiskom na srednju tipku za potvrdu. Osoba koja je ušla ima 15 s vremena da unese pravilnu kombinaciju. Ako se unese ispravna kombinacija zvuk upozorenja se isključuje. Ako se unese pogrešna kombinacija, sustav mora kratkim zvučnim signalom obavijestiti da je unesena kriva kombinacija i nastaviti sa kratkim zvučnim signalima za unos šifre. Nakon toga korisnik može pokušati unijeti novu kombinaciju. Ako korisnik ne unese ispravnu kombinaciju unutar 15s sustav mora aktivirati alarm koji se isključuje tek ako je unesena ispravnn kombinacija.

Reed prekidači i zvučnik povezani su na sustav preko GPIO sklopa (GPIO\_SNAPWIRE).

(OPREMA: E2LP ploča, dva HAA27 reed prekidača, zvučnik)

# Rješenje / Solution

U prostor između linija kopirati asemblerski kod Vašeg rješenja uz komentare (na engleskom jeziku). Uključite i kod potreban za provjeru rješenja. Prilikom kopiranja koristite opciju Paste -> Keep text only kako bi sačuvali izvorni format.

Copy your assembly code between lines. Include code required for verification of the solution. Please comment the code. Use Paste -> Keep text only when copying to preserve original formatting.

LEDCR EQU 0FFFFD000 ;LED diodas

LEDDATA EQU 0FFFFD004

ZVUCNIK\_CR EQU 0FFFFC100 ;GPIO Snapwire high –output

ZVUCNIK\_DR EQU 0FFFFC104

RTC1CR EQU 0FFFF0100 ;RTC1

RTC1LR EQU 0FFFF0104

RTC2CR EQU 0FFFF0200 ;RTC2

RTC2LR EQU 0FFFF0204

RTC2IACK EQU 0FFFF0208

RTC2END EQU 0FFFF020C

BUTTONC EQU 0FFFFE000 ;buttons

BUTTOND EQU 0FFFFE004

BUTTONIACK EQU 0FFFFE008

BUTTONEND EQU 0FFFFE00C

SENZORCR EQU 0FFFFC000 ;GPIO Snapwire low –input

SENZORDR EQU 0FFFFC004

SENZORIACK EQU 0FFFFC008

SENZOREND EQU 0FFFFC00C

MILISEKUNDE EQU %D 6750 ;f = 6,75MHz t=1 ms= 10^(-3)s-->

;1/t=1000 --->fms = 6,75 \* 1000 (to CT1)

SEKUNDE EQU %D 1000 ;1 second = 1000 \* 1 milisecond

PETNAEST EQU %D 15000 ;15 seconds = 15000 \* 1 miliseconds (to CT2)

;------------------------------------------------------------------------- ORG 0

MOVE 1000, SP

JR GLAVNI

ORG 8

VEKTOR DW 500

;----------------------------------------------------------------------------------------------------------------------------------

ORG 0C ;NMI

PREKID PUSH R0

MOVE 0, R0

STORE R0, (RTC2IACK)

LOAD R0, (SVIRAJ)

CMP R0, 0 ;compare sound with 0

JR\_EQ END ;if a sound is off –the end

; (password has been correct)

MOVE 100, R0

STORE R0, (FREKV) ;set up the frequency of alarm

MOVE 1, R0

STORE R0, (LEDDATA) ;make a dioda shine

END MOVE %B 110, R0

STORE R0, (RTC2CR) ;disable counting for RTC2

MOVE %B 100, R0

STORE R0, (RTC1CR) ;disable counting for RTC1

STORE R0, (RTC2END)

POP R0

RETN

;-------------------------------------------------------------------------

GLAVNI MOVE MEMORIJA, R0 ;move the address of MEMORIJA to ADRESA

; (ADRESA is pointer to MEMORIJA)

STORE R0, (ADRESA)

LOAD R0, (LED\_INIT) ;initialization of LED diodas

STORE R0, (LEDCR)

MOVE %B 10, R0 ;initialization of speaker

STORE R0, (ZVUCNIK\_CR)

LOAD R0, (MASKA1SENZOR) ;initialization of sensors

;("doors") - interrupt disabled

STORE R0, (SENZORCR)

LOAD R1, (MASKABUTTON) ;initialization of buttons

STORE R1, (BUTTONC)

MOVE PETNAEST, R0 ;initialization of RTC2

STORE R0, (RTC2LR) ;time

LOAD R0, (RTC2\_INIT)

STORE R0, (RTC2CR) ;settings

MOVE MILISEKUNDE, R0 ;initialization of RTC1

STORE R0, (RTC1LR) ;time

LOAD R0, (RTC1START)

STORE R0, (RTC1CR) ;settings

MOVE %B 10000, SR ;INT interrupts enabled

POSTAVILOZINKU LOAD R0, (POSTAVISIFRU) ;check if password is set up

; (at the beginning)

CMP R0, 1

JR\_NE POSTAVILOZINKU ;if it's not, wait while it is

CALL SOUND ;make a sound

;(to mark the beginning of 10s)

CALL DESETSEKUNDI ;wait 10s

CALL SOUND ;make a sound (to mark the end of 10s)

LOAD R0, (MASKA2SENZOR) ;initialization of sensors ("doors") –

; interrupt enabled

STORE R0, (SENZORCR)

MOVE 0, R0 ;do not make any sound

STORE R0, (SVIRAJ)

;-------------------------------------------------------------------------

PETLJA LOAD R1, (SVIRAJ) ;check for sound

STORE R1, (LEDDATA) ;if SVIRAJ=1 LED dioda shines, else it

;doesn't

CMP R1, 0

JR\_EQ PETLJA ;if sound should be turned off,

;continue to loop

CALL SOUND ;else make a sound

LOAD R1, (PAUZA) ;if PAUZA=1 make a short sound

;(if wrong password is entered)

CMP R1, 1

JR\_NE PRESKOCI ;else don't do that (continue to loop)

LOAD R2, (FREKV) ;save the old frequency

MOVE %D 100, R1

STORE R1, (FREKV) ;new frequency for a short sound

CALL SOUND ;make a sound two times(it's very

;fast so we'll hear only one)

CALL SOUND

MOVE 0, R1

STORE R1, (PAUZA) ;set PAUZA=0

STORE R2, (FREKV) ;return the old frequency

PRESKOCI JR PETLJA ;continue doing loop "forever"

;------------------------------------------------------------------------

ORG 500 ;INT

LOAD R0, (SENZORIACK) ;if one of the sensors (2 doors)

;has generated interrupt

;continue to SENZOR

CMP R0, 1

JR\_NE BUTTON ;else the button has generated

;interrupt, continue to BUTTON

SENZOR STORE R0, (SENZORIACK)

LOAD R0, (USAO) ;check if person has already

;opened the door

CMP R0, 1

JR\_EQ KRAJ ;if a person has already opened

;the door, finish

MOVE 1, R0

STORE R0, (SVIRAJ) ;activate a sound

STORE R0, (USAO) ;mark that a person has opened the door

;SET UP THE TIMER (15 seconds)

MOVE PETNAEST, R0 ;initialization of RTC2

STORE R0, (RTC2LR)

MOVE MILISEKUNDE, R0 ;initialization of RTC1

STORE R0, (RTC1LR)

MOVE %B 111, R0 ;RTC2--NMI, interrupt enabled, start counting

STORE R0, (RTC2CR)

MOVE %B 101, R0

STORE R0, (RTC1CR) ;RTC1--101--NMI,interrupt disabled,

;start counting

KRAJ STORE R0, (SENZOREND)

RETI

;-------------------------------------------------------------------------

BUTTON STORE R0, (BUTTONIACK)

LOAD R2, (BROJAC) ;the counter

LOAD R3, (ADRESA) ;the memory address

LOAD R4, (BUTTOND) ;load the data (pushed button)

CMP R4, %B 100 ;is it the center button?

JR\_EQ CENTER ;if it's not, continue

LOAD R5, (R3) ;load the pasword (one button)

;from memory

ADD R3, 4, R3 ;increase the memory adress

STORE R3, (ADRESA) ;store the new adress to the memory

CMP R4, R5 ;compare the pushed button

;and the password

JR\_NE RAZLICITI

ADD R2, 1, R2 ;if they are the same,

;increase the counter

JR IZLAZ

RAZLICITI MOVE MEMORIJA, R3 ;reset the address to the initial one

STORE R3, (ADRESA) ;and store it to the memory

MOVE 0, R2 ;reset the counter

JR IZLAZ ;exit (the password isn't complete)

CENTER MOVE MEMORIJA, R3 ;reset the address to the initial one

STORE R3, (ADRESA)

CMP R2, 4 ;if it's the center button,

;compare a counter with 4

JR\_EQ NASTAVI ;if the counter is 4 finish

;(stop the sound or alarm)

MOVE 1, R2 ;for making a short sound - PAUZA=1

STORE R2, (PAUZA)

MOVE 0, R2 ;reset the counter

JR IZLAZ ;exit (the password isn't correct)

NASTAVI MOVE 500, R2 ;set up the old frequency (because

;the alarm has different frequency)

STORE R2, (FREKV)

LOAD R2, (POSTAVISIFRU) ;if password isn't set up at

;(the beginning), set it up

CMP R2, 0

JR\_NE CONT ;else continue

MOVE 1, R2 ;set up the password

STORE R2, (POSTAVISIFRU)

MOVE 0, R2 ;reset the counter

STORE R2, (SVIRAJ) ;disable a sound (0-disable)

STORE R2, (USAO) ;USAO=0

JR IZLAZ

CONT MOVE 0, R2 ;reset the counter

STORE R2, (RTC2CR) ;disable counting for RTC2

STORE R2, (RTC1CR) ;disable counting for RTC1

STORE R2, (SVIRAJ) ;disable a sound (0-disable)

STORE R2, (USAO) ;a new person can enter the house

;(doors are closed)

STORE R2, (RTC1LR) ;reset the counter of RTC1LR

STORE R2, (RTC2LR) ;reset the counter of RTC2LR

JR IZLAZ

IZLAZ STORE R2, (BROJAC) ;store the counter to the memory

STORE R0, (BUTTONEND)

RETI

;-------------------------------------------------------------------------

;----WAIT 10 SECONDS---

DESETSEKUNDI LOAD R0, (BROJAC10)

OR R0, R0, R0 ; 1 period

SUB R0, 1, R0 ; 1 period

STORE R0, (BROJAC10) ; 2 periods

JR\_NZ DESETSEKUNDI ; 2 periods (at the end 1 period)

RET

;--------------------------------------------------------------------------

;----MAKE A SOUND----

SOUND PUSH R0

MOVE %D 100 , R6 ;a constant

PET1 MOVE 1, R0

STORE R0, (ZVUCNIK\_DR) ;send 1 to speaker

CALL CEKAJ1 ;wait

MOVE 0, R0

STORE R0, (ZVUCNIK\_DR) ;send 0 to speaker

CALL CEKAJ1 ;wait

SUB R6 , 1 , R6 ;reduce a constant

JR\_NZ PET1 ;if it's 0 finish

POP R0

CEKAJ1 PUSH R1

LOAD R1, (FREKV) ;load frequency

LOOP1 SUB R1, 1, R1 ;reduce a frequency

JR\_NZ LOOP1 ;if it's 0 finish

POP R1

RET

;--------------------------------------------------------------------------

;the password

MEMORIJA DW %B 10 ;down

DW %B 1 ;right

DW %B 10 ;down

DW %B 10 ;down

ADRESA DW 0 ;the pointer to MEMORIJA

POSTAVISIFRU DW 0 ; =1 if password is set up,

; else =0

USAO DW 0 ; =1 if person has opened

; the door, else =0

BROJAC DW 0 ;the counter

SVIRAJ DW 0 ; =1 if a sounf is enabled,

;else =0

FREKV DW 500 ;the frequency

PAUZA DW 0 ; if =1 - make a short sound

;(for wrong password)

MASKABUTTON DW %B 111110001111100000111

;buttons - ACTIVE=1, MASK- 5 buttons, OR, INT, interrupt enabled, MODE: cheking out bits

MASKA1SENZOR DW %B 000000000000000011

;sensors - MODE:cheking out bits, interrupt disabled

MASKA2SENZOR DW %B 000000001100000111

;sensors - ACTIVE=0, MASKA- 2 doors,OR,INT,interrupt enabled, MODE: cheking out bits

LED\_INIT DW %B 10 ;LED diodas - setting up bits

RTC1START DW %B 101 ;RTC1 - only counts

RTC2\_INIT DW %B 111 ;RTC2 - counts and generates interrupt

BROJAC10 DW %D 2200000 ;a constant for DESETSEKUNDI

# Ispitivanje rješenja / Solution Verification

U prostor između linija objasnite kako se provjerava ispravnost rješenja (na hrvatskom ili na engleskom jeziku):

Write detailed explanation how is the solution verified:

Na početku korisnik upisuje šifru pomoću tipki (potvrđuje ju pritiskom na srednju tipku). Dok ne upiše ispravnu šifru, neće se čuti zvuk koji označava početak 10 sekundi. U tih 10 sekundi vlasnik kuće mora napustiti kuću i zatvoriti vrata, odnosno tada se otvaranjem vrata (pomicanjem reed prekidača) ne aktivira nikakav zvuk (vrata ne postavljaju prekid). Nakon 10 sekundi još se jednom čuje zvuk koji označava da je prošlo zadano vrijeme.

Ako netko otvori vrata (ako se razmaknu reed prekidači), aktiviraju se zvučni signali te brojilo počne brojati 15 sekundi. Ako se upiše neispravna šifra, čuje se kratki zvučni signal visoke frekvencije. Ako se upiše ispravna lozinka, zvučni se signali isključe te se brojilo koje broji 15 sekundi zaustavi i resetira (ne aktivira se alarm). Ako osoba u 15 sekundi ne upiše ispravnu šifru, aktivira se alarm koji ima drukčiju frekvenciju (višu od one od prethodnih zvučnih signala) te koji se gasi tek kad se upiše ispravna šifra. Dok se čuju zvučni signali pomicanjem reed prekidača (otvaranjem vrata) ne događa se novi prekid reed prekidača (dakle brojilo koje broji 15 sekundi se ne resetira). Kada su zvučni signali isključeni, novim otvaranjem vrata ponavlja se process.