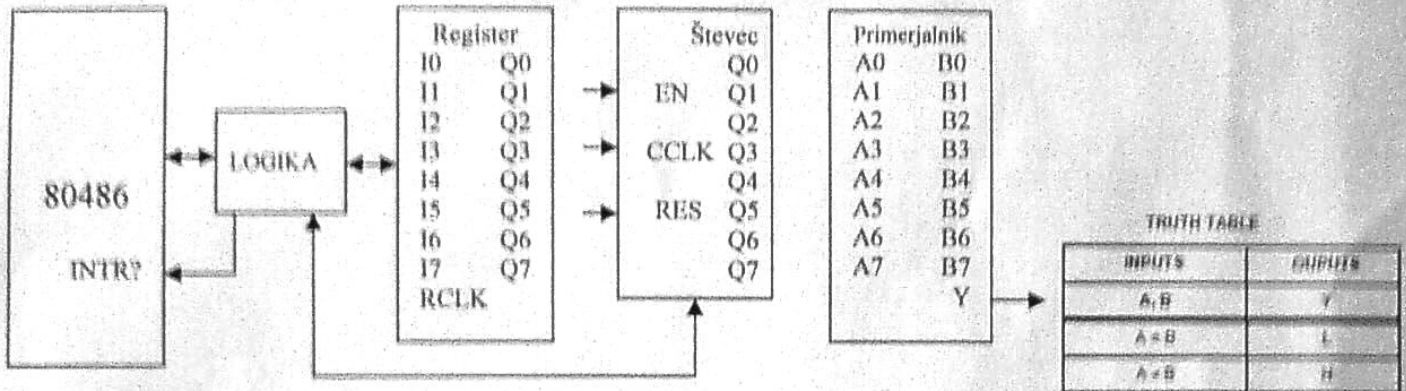


1. Za mikroprocesor 80486 narišite logiko ki bo generirala prekinitveno zahtevo, ko števec zunanjih dogodkov doseže določeno vrednost. Za štetje zunanjih dogodkov uporabite 8-bitni števec na sliki. Če je vhod EN v visokem stanju, števec šteje impulze na vhodu CCLK. Vhod CCLK je že vezan na nek zunanji signal in ga ni potrebno dodatno povezovati. Ob visokem napetostnem nivoju na vhodu RES pa se vrednost števca (stanje izhodov Q0-Q7) postavi na 0 in to ne glede na stanje vhodov CCLK in EN. Vrednost, s katero se stanje števca primerja, naj bo mogoče programsko določiti z vpisom v 8-bitni register, ki se nahaja na V/I naslovu FF03. Stanje vhodov I0-17 se v register vpiše in pojavi na izhodih Q0-Q7 ob pozitivni fronti na vhodu RCLK. Ob pisanju v register naj se vrednost števca postavi na 0. Za primerjanje vrednosti registra in vrednosti števca uporabite primerjalnik na sliki. Ko vrednost števca postane enaka vrednosti registra, naj se štetje ustavi, logika pa naj na prekinitveni vhod 80486 postavi prekinitveno zahtevo (logike za pošiljanje prekinitvenega vektorja ni potrebno narisati).



2. a) Za mikrokrmilnik ARM (AT91SAM9260) napišite prekinitveno servisni program, ki omogoča sprejem podatkov s pomočjo USART vmesnika. Ob vsaki prekinitvi sprejmemo po en znak. Najprej mora program preveriti ali je do prekinitve prišlo zaradi sprejema novega znaka. Če nov znak ni na voljo, mora program v 8-bitno spremenljivko **ERROR** vpisati vrednost 1 in se ustrezno zaključiti, če pa znak je na voljo, pa mora sprejeti znak zapisati v krožni izravnalnik **BUFFER** velikosti 32 bajtov na mesto, ki ga določa vrednost 8-bitne spremenljivke **BUF_HEAD**. Vrednost spremenljivke **BUF_HEAD** mora tudi ustrezno popraviti, da bo pravilno kazala položaj za sprejem naslednjega znaka. Dodatna pojasnila: Dostop do podatkov v krožnem izravnalniku velikosti **N** poteka tako, da zaporedno dostopamo do podatkov od naslova 0 do **N-1**, naslednji dostop pa spet poteka na naslovu 0. Poleg polja za hranjenje podatkov velikosti **N**, imamo v podatkovni strukturi krožnega izravnalnika tudi spremenljivko, ki hrani trenutni naslov, podatka do katerega bomo dostopali pri naslednjem dostopu.

```
.equ USART_BASE, 0xFFFFB000    /* USART0 Base Address */
.equ USART_CSR, 0x14           /* Channel Status Register */
.equ USART_RHR, 0x18          /* Receiver Holding Register */

.equ AIC_BASE, 0xFFFFF000     /* AIC Base Address */
.equ AIC_EOICR, 0x130        /* End of Interrupt Command Register */
```

```
BUFFER:      .space      32
BUF_HEAD:    .byte       0
ERROR:       .byte       0
```

7	6	5	4	3	2	1	0
FARE	FRAME	OVRE	ENDTX	ENDRX	RXBK	TXRDY	RXRDY

• **RXRDY: Receiver Ready**

0: No complete character has been received since the last read of **US_RHR** or the receiver is disabled. If characters were being received when the receiver was disabled, **RXRDY** changes to 1 when the receiver is enabled.

1: At least one complete character has been received and **US_RHR** has not yet been read.

Prikaz spodnjih 8 bitov registra **USART_CSR**.

b) Za mikrokontroler 68HC11 napišite prekinitevno servisni program, ki omogoča sprejem podatkov s pomočjo USART vmesnika, ki je priključen na mikrokontroler. Ob vsaki prekinitvi sprejmemo po en znak. Najprej mora program preveriti ali je do prekinitve prišlo zaradi sprejema novega znaka. Če nov znak ni na voljo, mora program v 8-bitno spremenljivko ERROR vpisati vrednost 1 in se ustrezno zaključiti, če pa znak je na voljo, pa mora sprejeti znak zapisati v krožni izravnalnik BUFFER velikosti 32 bajtov na mesto, ki ga določa vrednost 8-bitne spremenljivke BUF_HEAD. Program mora vrednost spremenljivke BUF_HEAD tudi ustrezno popraviti, da bo pravilno kazala položaj za sprejem naslednjega znaka. Dodatna pojasnila: Dostop do podatkov v krožnem izravnalniku velikosti N poteka tako, da zaporedno dostopamo do podatkov v polju od naslova 0 do N-1, naslednji dostop pa spet poteka na naslovu 0. Poleg polja za hranjenje podatkov velikosti N imamo v podatkovni strukturi krožnega izravnalnika tudi spremenljivko, ki hrani trenutni naslov podatka, do katerega bomo dostopali pri naslednjem dostopu.

```

USART_DATA    EQU    $A000    /* USART data holding register*/
USART_STATUS  EQU    $A001    /* USART Status Register */

BUFFER:       RMB        32
BUF_HEAD:     FCB        0
ERROR:        FCB        0

```

7	6	5	4	3	2	1	0
PARF	FRAME	OVRE	ENDTX	ENDRX	RXBK	TXRDY	RXRDY

RXRDY: Receiver Ready

0: No complete character has been received since the last read of USART_DATA.

1: At least one complete character has been received and USART_DATA has not yet been read.

Prikaz registra USART_STATUS.

3. Izračunajte (z zaokrožitvijo) število sektorjev na notranji in zunanji sledi diska z naslednjimi podatki: hitrost vrtenja diska je 5454 obratov na minuto, notranja hitrost prenosa pa je med 11 in 21 MB na sekundo. Predpostavite, da je velikost sektorja 512 bajtov in gostota zapisa enaka za cel disk.

237,557

TOČKE: 1 - 40, 2a ali 2b - 40, 3 - 20; Čas reševanja: 75 minut.

Rezultati bodo objavljeni v petek 11.6.2010, ustni izpiti bodo od ponedeljka 14.6.2007 dalje.