

# Integrirani računarski sistemi

## Serijska komunikacija

Odsek za elektroniku

Univerzitet u Beogradu - Elektrotehnički fakultet

poslednja izmena 3. mart 2022



## 1 UART

- Principi
  - Primer – UART echo
- Prekidi
  - Primer – UART terminal
- Paketski prenos
  - Primer – UART interrupt echo
- UART multiprocesorska komunikacija

## 2 SPI

## 3 I<sup>2</sup>C



## 1 UART

- Principi
  - Primer – UART echo
- Prekidi
  - Primer – UART terminal
- Paketski prenos
  - Primer – UART interrupt echo
- UART multiprocesorska komunikacija

## 2 SPI

## 3 I<sup>2</sup>C



## 1 UART

- Principi

- Primer – UART echo

- Prekidi

- Primer – UART terminal

- Paketski prenos

- Primer – UART interrupt echo

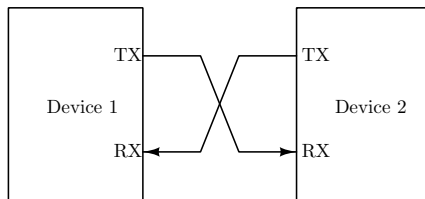
- UART multiprocesorska komunikacija

## 2 SPI

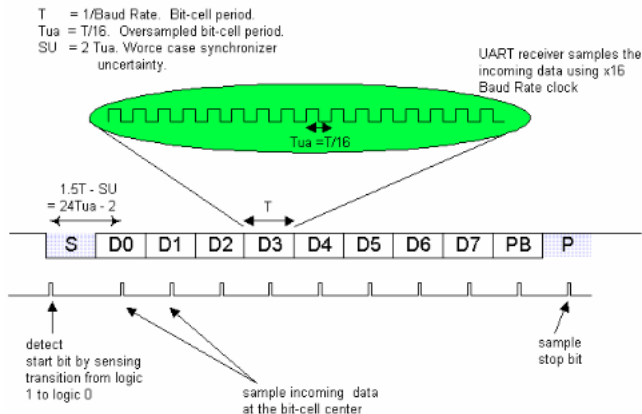
## 3 I<sup>2</sup>C



# Serijska komunikacija



# Format prenosa podataka



# USCI - Universal Serial Communication Interface

USCI moduli podržavaju više serijskih protokola

USCI\_A $x$  ( $x = 0 - 1$ )

- UART
- Oblikovanje signala za IrDA komunikaciju
- Automatsko detektovanje brzine prenosa za LIN komunikaciju
- SPI

USCI\_B $x$  ( $x = 0 - 1$ )

- I<sup>2</sup>C
- SPI



Bira se sa `UCAxCTL0.UCSYNC=0`

Podatak dužine 7 ili 8 bita i bit parnosti

Nezavisan rad prijemnika i predajnika

Baferisani prijemni i predajni registri

Ugrađena podrška za multiprocesorsku komunikaciju

Odvojeni prekidi za prijem i predaju





Mogućnost buđenja iz LPMx moda na startnu ivicu signala na ulazu u prijemnik

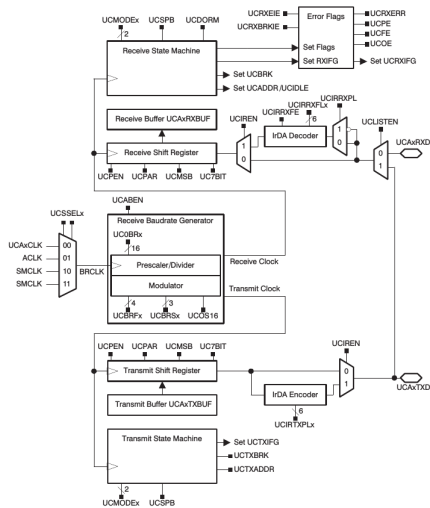
Mogućnost podešavanja brzine prenosa u širokom opsegu vrednosti i sa velikom rezolucijom

Hardverska detekcija grešaka u prenosu

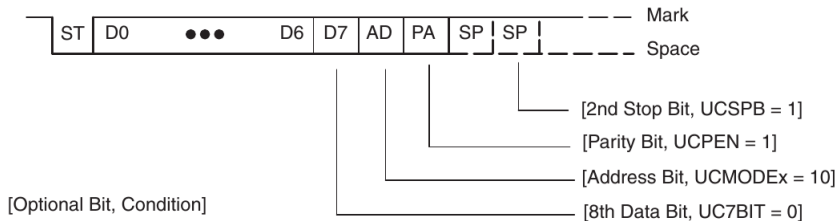
Prekid za prijemnik i predajnik



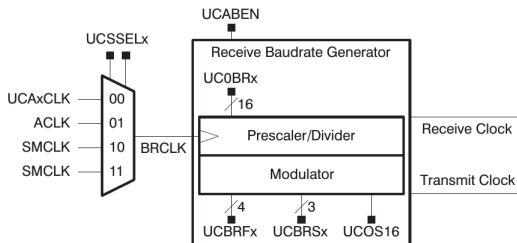
# USCI - UART mod



# Format prenosa podataka



# Baud rate generator



Podržana su dva režima rada koji se biraju sa UCOS16

Faktor  $N$  kojim se deli učestanost  $BRCLK$  u cilju dobijanja željene bitske učestanosti se dobija kao

$$N = \frac{f_{BRCLK}}{Baudrate}$$

## Low-Frequency režim (UCOS16=0)

Omogućava generisanje željenog *baud rate*-a na osnovu izvora takta niske učestanosti - smanjena potrošnja, mada mogu da se koriste i izvori visoke učestanosti

Maksimalan *baud rate* je jedna trećina učestanosti takta BRCLK

### Podešavanje

$$UCBR_x = \text{INT}(N)$$

$$UCBR_{Sx} = \text{round} [(N - \text{INT}(N)) \times 8]$$



# Low-Frequency režim (UCOS16=0)

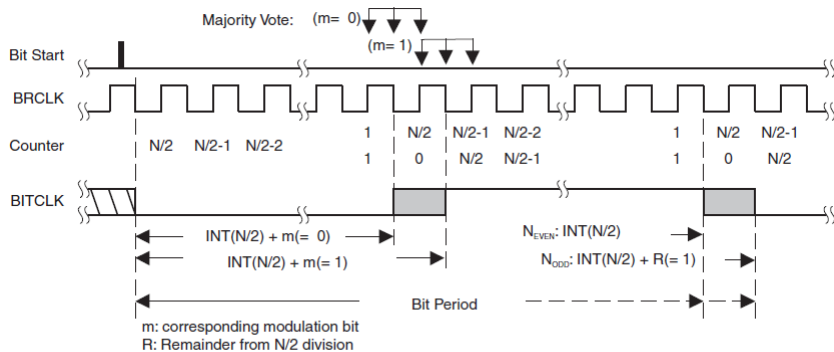


Figure 36-10. BITCLK Baud-Rate Timing With UCOS16 = 0



## Oversampling režim (UCOS16=1)

Omogućava generisanje željenog *baud rate*-a na osnovu izvora takta visoke učestanosti

Vrši se *oversampling* sa 16 puta većom učestanosti

Maksimalan *baud rate* je jedna šesnaestina učestanosti takta BRCLK

### Podešavanje

$$\text{UCBR}_x = \text{INT}(N/16)$$

$$\text{UCBRF}_x = \text{round} [(N/16 - \text{INT}(N/16)) \times 16]$$



# Standardne brzine i podešavanja UCBRx i UCBSx

**Table 36-4. Commonly Used Baud Rates, Settings, and Errors, UCOS16 = 0**

BRCLK Frequency (Hz)	Baud Rate (baud)	UCBRx	UCBSx	UCBRFx	Maximum TX Error (%)		Maximum RX Error (%)	
32 768	1200	27	2	0	-2.8	1.4	-5.9	2.0
32 768	2400	13	6	0	-4.8	6.0	-9.7	8.3
32 768	4800	6	7	0	-12.1	5.7	-13.4	19.0
32 768	9600	3	3	0	-21.1	15.2	-44.3	21.3
1 000 000	9600	104	1	0	-0.5	0.6	-0.9	1.2
1 000 000	19200	52	0	0	-1.8	0	-2.6	0.9
1 000 000	38400	26	0	0	-1.8	0	-3.6	1.8
1 000 000	57600	17	3	0	-2.1	4.8	-6.8	5.8
1 000 000	115200	8	6	0	-7.8	6.4	-9.7	16.1
1 048 576	9600	109	2	0	-0.2	0.7	-1.0	0.8
1 048 576	19200	54	5	0	-1.1	1.0	-1.5	2.5
1 048 576	38400	27	2	0	-2.8	1.4	-5.9	2.0
1 048 576	57600	18	1	0	-4.6	3.3	-6.8	6.6
1 048 576	115200	9	1	0	-1.1	10.7	-11.5	11.3
4 000 000	9600	416	6	0	-0.2	0.2	-0.2	0.4
4 000 000	19200	208	3	0	-0.2	0.5	-0.3	0.8
4 000 000	38400	104	1	0	-0.5	0.6	-0.9	1.2
4 000 000	57600	69	4	0	-0.6	0.8	-1.8	1.1
4 000 000	115200	34	6	0	-2.1	0.6	-2.5	3.1



## 1 UART

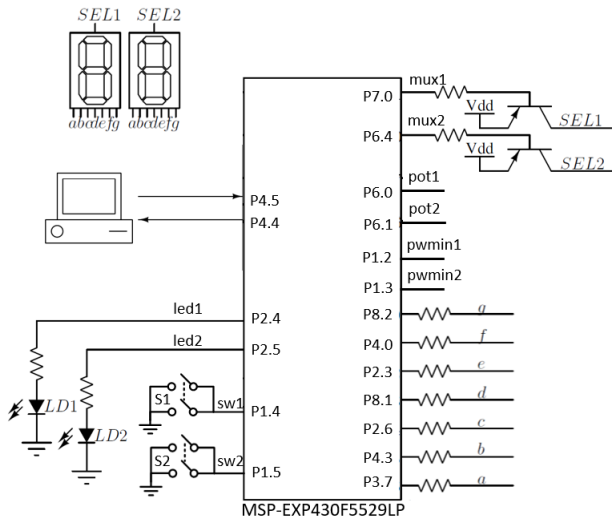
- Principi
  - Primer – UART echo
- Prekidi
  - Primer – UART terminal
- Paketski prenos
  - Primer – UART interrupt echo
- UART multiprocesorska komunikacija

## 2 SPI

## 3 I<sup>2</sup>C



# Opis hardvera



# Terminalska aplikacija

Na računaru je potrebno koristiti terminalsku aplikaciju za povezivanje putem serijske veze

CCS ima integrisanu terminalsku aplikaciju koja se pokreće iz **View -> Terminal**

U okviru aplikacije potrebno je podesiti

- port računara putem kojeg se ostvaruje serijska veza
  - u *Device manager*-u videti koji je COM port u pitanju
- brzinu komunikacije (*baud rate*)
- bit parnosti
- broj stop bita
- *flow control*

# Zadatak – UART echo

## Zadatak

Napisati program za razvojni sistem koji obezbeđuje serijsku vezu mikrokontrolera sa PC računarom. Brzina komunikacije treba da bude 9600 bps, bez bita parnosti i sa jednim stop bitom (9600 8N1). Omogućiti da se po prijemu podatka sa računara (ASCII kod odgovarajućeg karaktera) računaru pošalje taj podatak uvećan za 1 po modulu 255.

## Napomene

Koristiti poliranje za dobijanje informacije o pristiglom/poslatom podatku

## Rešenje

uart-echo



---

**NOTE: Initializing or reconfiguring the USCI module**

The recommended USCI initialization/reconfiguration process is:

1. Set UCSWRST (BIS.B  
#UCSWRST, &UCAxCTL1).
  2. Initialize all USCI registers with UCSWRST = 1 (including UCAxCTL1).
  3. Configure ports.
  4. Clear UCSWRST via software (BIC.B  
#UCSWRST, &UCAxCTL1).
  5. Enable interrupts (optional) via UCRXIE and/or UCTXIE.
- 



## 1 UART

- Principi
  - Primer – UART echo
- Prekidi
  - Primer – UART terminal
- Paketski prenos
  - Primer – UART interrupt echo
- UART multiprocesorska komunikacija

## 2 SPI

## 3 I<sup>2</sup>C



## Drugi pristup - UART prekidi

USCI poseduje jedan prekidni vektor koji se koristi i za prijemnik i za predajnik

Prekid prijemnika se događa kada je podatak primljen i upisan u `UCAxRXBUF`

Prekid predajnika se događa kada je sadržaj predajnog registra `UCAxTXBUF` prebačen u izlazni pomerački registar i može se upisati novi podatak

Registar `UCAxIV` sadrži informaciju o aktivnom zahtevu za prekid najvišeg nivoa

- prekid prijemnika ima viši prioritet od prekida predajnika



# Primer prekidne rutine

```
USCI_UART_ISR
    ADD    &UCA0IV, PC    ; Add offset to jump table
    RETI    ; Vector 0: No interrupt
    JMP    RXIFG_ISR      ; Vector 2: RXIFG
TXIFG_ISR    ; Vector 4: TXIFG
    ...    ; Task starts here
    RETI    ; Return
RXIFG_ISR    ; Vector 2
    ...    ; Task starts here
    RETI    ; Return
```





## 1 UART

- Principi
  - Primer – UART echo
- Prekidi
  - Primer – UART terminal
- Paketski prenos
  - Primer – UART interrupt echo
- UART multiprocesorska komunikacija

## 2 SPI

## 3 I<sup>2</sup>C



# Zadatak – UART terminal

## Zadatak

Napisati program za razvojni sistem koji obezbeđuje serijsku vezu mikrokontrolera sa PC računarom. Brzina komunikacije treba da bude 9600 bps, bez bita parnosti i sa jednim stop bitom (9600 8N1). Pritiskom na taster **S1** potrebno je generisati serijsko slanje podatka koji se nalazi u promenljivoj **data**. Sa druge strane, numerički jednocifreni podatak primljen preko serijske veze treba snimiti u promenljivu **data** i ispisati ga na sedmosegmentnom LED displeju.

## Rešenje

- **uart-terminal-c** – prekidne rutine porta i UART-a implementirane u C-u
- **uart-terminal-mix** – prekidne rutine porta i UART-a implementirane u assembleru

## 1 UART

- Principi
  - Primer – UART echo
- Prekidi
  - Primer – UART terminal
- Paketski prenos
  - Primer – UART interrupt echo
- UART multiprocesorska komunikacija

## 2 SPI

## 3 I<sup>2</sup>C

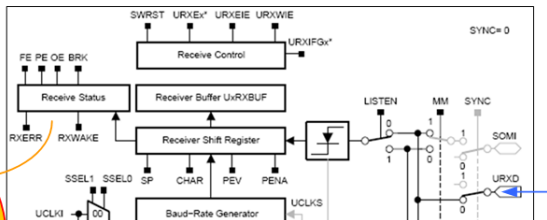


# Interrupt driven receiving

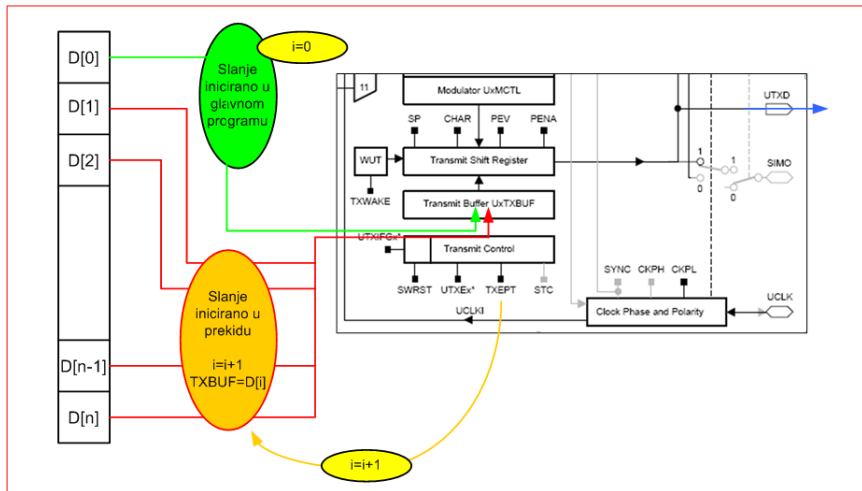


Prijem prekida

```
i=i+1
D[i]=RXBUF
If(RXBUF==0xFF) i=0
If(i=n)
MSG_ARRIVED
```



# Interrupt driven transmitting



## 1 UART

- Principi
  - Primer – UART echo
- Prekidi
  - Primer – UART terminal
- Paketski prenos
  - Primer – UART interrupt echo
- UART multiprocesorska komunikacija

## 2 SPI

## 3 I<sup>2</sup>C



# Zadatak – UART interrupt echo

## Zadatak

Napisati program za razvojni sistem koji obezbeđuje serijsku vezu mikrokontrolera sa PC računarem. Brzina komunikacije treba da bude 38400 bps, bez bita parnosti i sa jednim stop bitom (38400 8N1). Omogućiti da se na prijem svaka 4 karaktera sa računara ti karakteri pošalju računaru u obrnutom redosledu od onog u kojem su stigli.

## Rešenje

uart-interrupt-echo



## 1 UART

- Principi
  - Primer – UART echo
- Prekidi
  - Primer – UART terminal
- Paketski prenos
  - Primer – UART interrupt echo
- UART multiprocesorska komunikacija

## 2 SPI

## 3 I<sup>2</sup>C





# Multiprocesorski mod

UART daje mogućnost da se iz serijskog niza karaktera izdvajaju adrese i podaci

Na taj način se omogućava vezivanje više uređaja (procesora) na serijsku magistralu

Dva načina multiprocesorske komunikacije

- „idle line” protokol
- protokol sa bitom markerom adrese



## „idle line” protokol 1/3

Jednom serijskom porukom se smatra niz karaktera koji se šalje u kontinuitetu tako da je vremensko rastojanje između dva karaktera manje od deset perioda serijskog takta

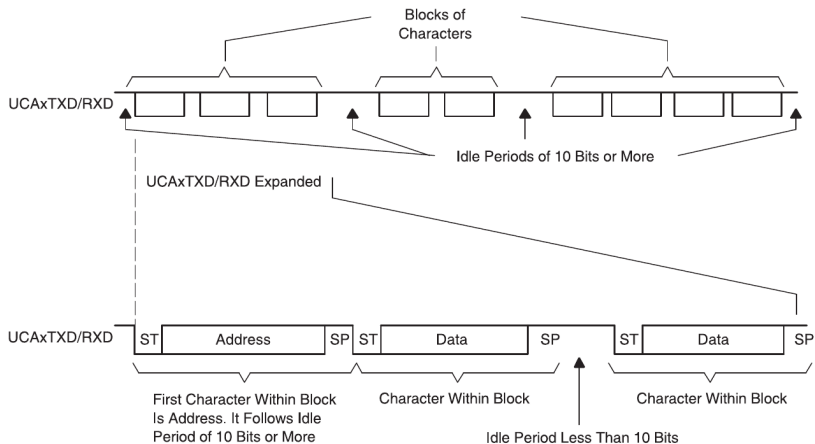
Prvi karakter u nizu predstavlja adresu

Niz karaktera koji slede su podaci

Svaki vremenski razmak veći od deset perioda serijskog takta smatra se krajem poruke



## „idle line” protokol 2/3



Predajnik automatski generiše idle line stanje kada se setuje odgovarajući bit u kontrolnom registru

Prijemnik može biti podešen da izaziva prekid samo po prijemu karaktera adrese, a da ostale karaktere ignoriše

Po prijemu odgovarajuće adrese u prekidnoj rutini se rekonfiguriše prijemnik za prijem podataka



# Multiprocesorski mod sa adresnim bitom 1/2

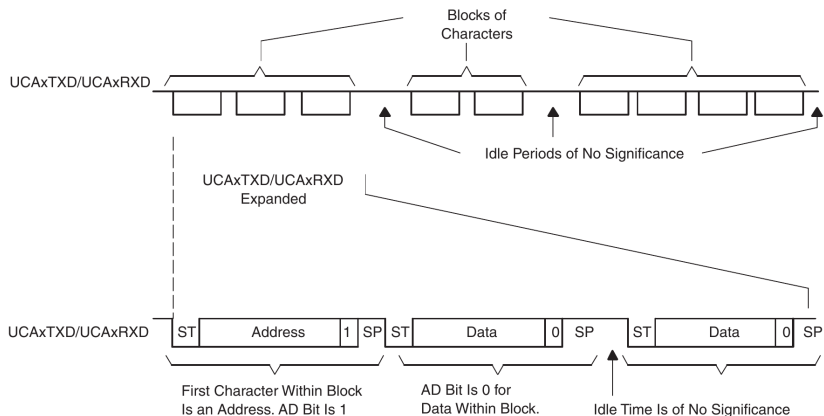
U okviru serijskog podatka koji sadrži 7 ili 8 bita može se dodati i deveti bit koji služi kao marker adrese

Kao i kod „idle line” moda prijemnik može biti podešen da izaziva prekid samo po prijemu karaktera adrese, a da ostale karaktere ignoriše

Po prijemu odgovarajuće adrese u prekidnoj rutini se rekonfiguriše prijemnik za prijem podataka



# Multiprocesorski mod sa adresnim bitom 2/2



## Prijemnik automatski prepoznaje greške u prenosu i setuje odgovarajuće flegove

Error Condition	Error Flag	Description
Framing error	UCFE	A framing error occurs when a low stop bit is detected. When two stop bits are used, both stop bits are checked for framing error. When a framing error is detected, the UCFE bit is set.
Parity error	UCPE	A parity error is a mismatch between the number of 1s in a character and the value of the parity bit. When an address bit is included in the character, it is included in the parity calculation. When a parity error is detected, the UCPE bit is set.
Receive overrun	UCOE	An overrun error occurs when a character is loaded into UCAXRXBUF before the prior character has been read. When an overrun occurs, the UCOE bit is set.
Break condition	UCBRK	When not using automatic baud-rate detection, a break is detected when all data, parity, and stop bits are low. When a break condition is detected, the UCBRK bit is set. A break condition can also set the interrupt flag UCRXIFG if the break interrupt enable UCBRKIE bit is set.



## 1 UART

- Principi
  - Primer – UART echo
- Prekidi
  - Primer – UART terminal
- Paketski prenos
  - Primer – UART interrupt echo
- UART multiprocesorska komunikacija

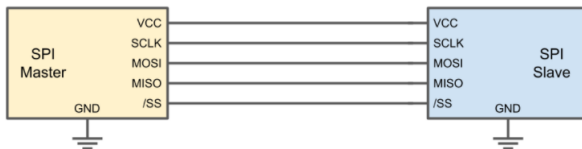
## 2 SPI

## 3 I<sup>2</sup>C

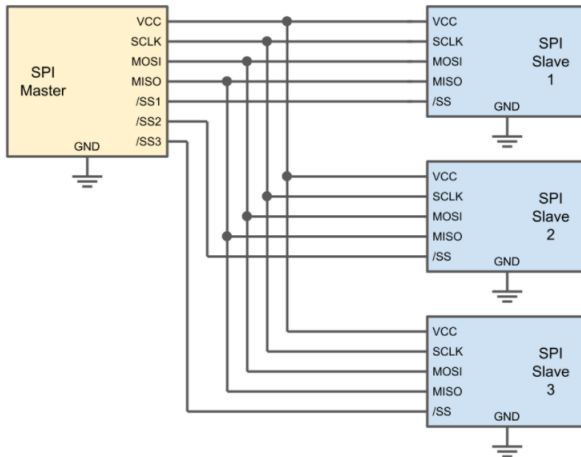




# SPI - serial peripheral interface



# SPI - several slaves



# SPI - serial peripheral interface

USCI\_Ax i USCI\_Bx podržavaju i sinhronu komunikaciju

Sinhrona serijska komunikacija

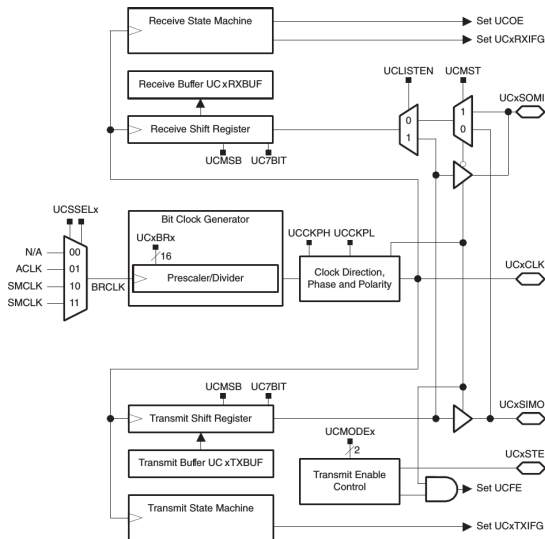
Fizički se ostvaruje korišćenjem tri pina: SIMO, SOMI, CLK

U slučaju vezivanja više master uređaja na magistralu koristi se i pin STE

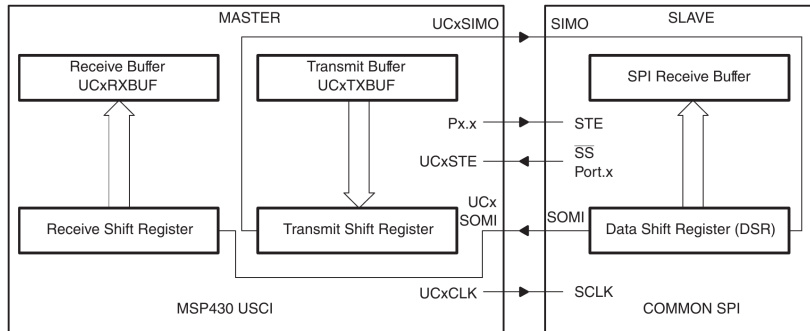
Podržava dužinu podatka od 7 ili 8 bita



# USCI u SPI modu



# SPI - MSP kao master 1/2

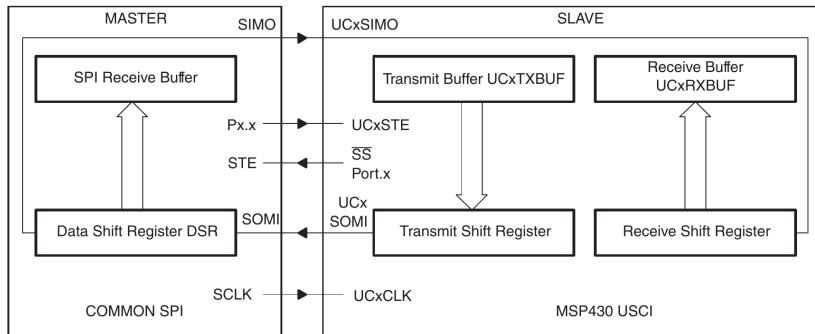


Proces slanja počinje upisom podatka u UCxTXBUF. Nakon toga podatak se prebacuje u pomerački registar i kreće slanje sa prvim bitom najveće težine. Uporedo sa slanjem podatka, od druge strane (slave) dobija se podatak koji se upisuje u prijemni pomerački registar. Po prijemu podatak se prebacuje iz pomeračkog registra u UCxRXBUF i izaziva se prekid

Operacije predaje i prijema su povezane i funkcionišu isključivo u paru



# SPI - MSP kao slave 1/2



## SPI - MSP kao slave 2/2

Podaci upisani u UCxTXBUF i prebačeni u pomerački registar pre pojave takta od strane mastera šalju se preko SOMI linije

Na suprotnu ivicu takta upisuju se podaci u ulazni pomerački registar

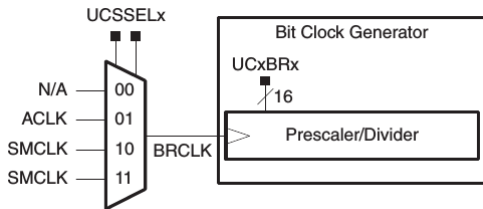
Po prijemu svih bita izaziva se prekid

Dok je STE pin na aktivnom nivou SPI funkcioniše normalno. STE na neaktivnom nivou zaustavlja prijem podataka, SOMI se konfiguriše kao ulazni pin





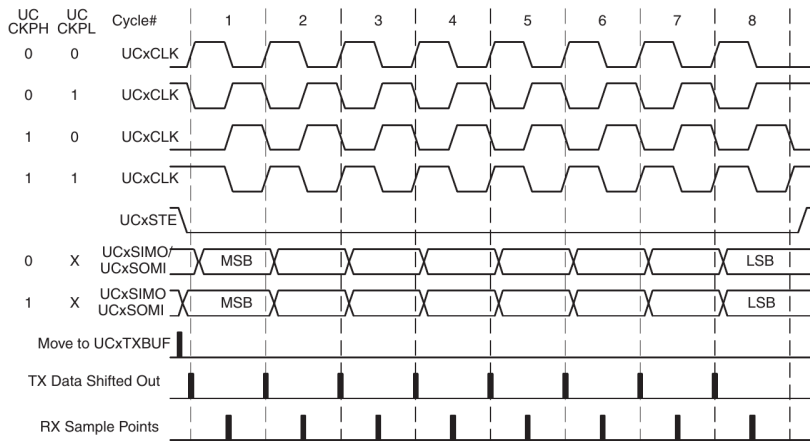
# Generator takta



$$f_{BitClock} = \frac{f_{BRCLK}}{UCBRx + 1}$$



# Kontrola takta



USCI poseduje jedan prekidni vektor koji se koristi i za prijemnik i za predajnik

Prekid prijemnika se događa kada je podatak primljen i upisan u `UCAxRXBUF`

Prekid predajnika se događa kada je sadržaj predajnog registra `UCAxTXBUF` prebačen u izlazni pomerački registar i može se upisati novi podatak

Registar `UCAxIV` sadrži informaciju o aktivnom zahtevu za prekid najvišeg nivoa

- prekid prijemnika ima viši prioritet od prekida predajnika



## 1 UART

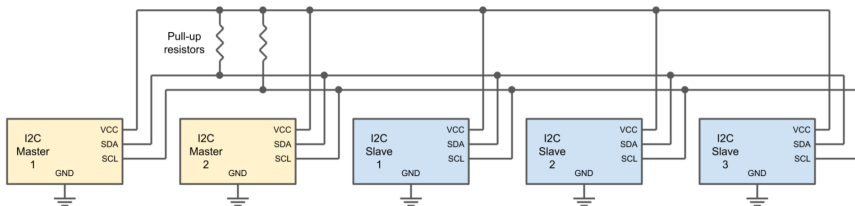
- Principi
  - Primer – UART echo
- Prekidi
  - Primer – UART terminal
- Paketski prenos
  - Primer – UART interrupt echo
- UART multiprocesorska komunikacija

## 2 SPI

## 3 I<sup>2</sup>C



# I2C - Inter-Integrated Circuit



# SPI vs I2C

	SPI	I2C
Pin drive	Push-pull	Open drain
Signal lines	4 (plus 1 for each additional peripheral)	2
Max speed	No limit (10-100 Mbps is common)	400 kbps in fast mode (3.4 Mbps is possible with high-speed mode)
No. of peripherals	Only limited by number of pins available for SS lines on master	112 with 7-bit addressing
Multi-master	No	Yes
Flow control	No	Yes



# I<sup>2</sup>C (inter-IC control) modul

USCI\_Bx podržava I<sup>2</sup>C komunikaciju

Zadovoljava Philips-ov I<sup>2</sup>C standard v2.1

Podržane 7-bitne ili 10-bitne adrese

Prenos bajta ili reči

Multi-master arbitracija

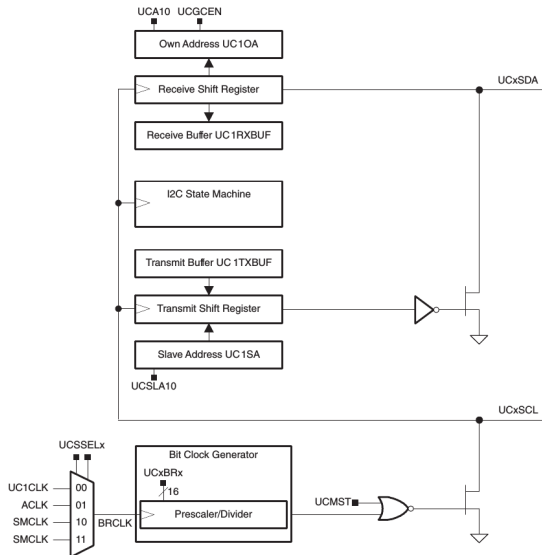
Master ili slave mod rada

Brzina prenosa do 400 kbps u fast modu

Mogućnost startovanja i iz LPM moda

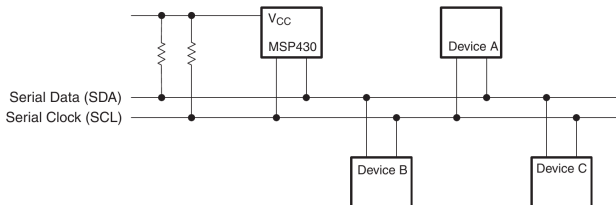


# USCI\_Bx u I<sup>2</sup>C modu





# I<sup>2</sup>C povezivanje na magistralu



Svaki uređaj na magistrali može da radi i kao predajnik i kao prijemnik

Svaki uređaj može takođe da bude ili master ili slave

Svatom uređaju je dodeljena jedinstvena 7-bitna ili 10-bitna adresa



## I<sup>2</sup>C signali na magistrali 1/2

Master je onaj uređaj koji diktira takt SCL. Svi ostali su slave-ovi.

Master započinje prenos obaranjem SDA signala dok je takt SCL na visokom nivou (START)

Prvi podatak u nizu je adresa, a posle nje idu podaci

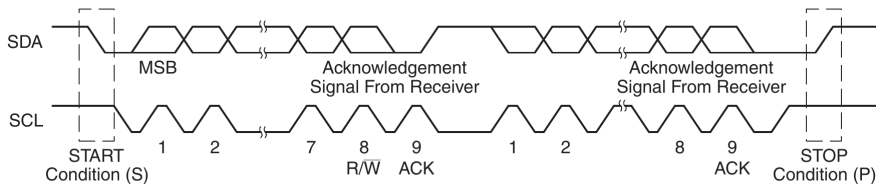
U zavisnosti od R/W bita master će u nastavku prenosa biti ili predajnik ili prijemnik



## I<sup>2</sup>C signali na magistrali 2/2

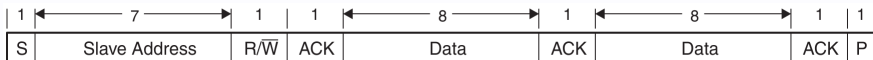
Slave posle primljene adrese ili prijemnik posle svakog primljenog podatka izdaje odziv **ACK** (prijemnik može da bude ili master ili slave). **ACK** se generiše kao držanje niskog nivoa na **SDA** liniji dok je takt aktivan

Po završetku prenosa master podiže nivo na **SDA** liniji dok je signal takta na visokom nivou (**STOP**)

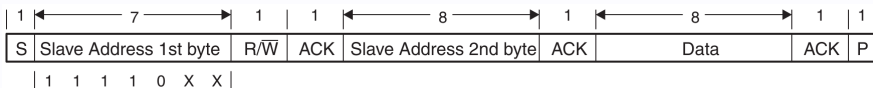


# Adresiranje

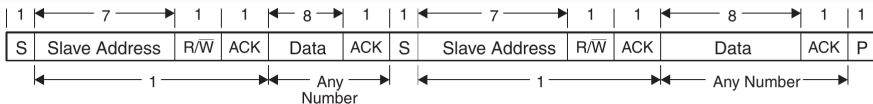
## 7-bitno adresiranje



## 10-bitno adresiranje



## Ponovljeni start



## I<sup>2</sup>C prekidi 1/2

Jedan prekidni vektor je zajednički za sve izvore prekida

Postoji šest različitih izvora prekida koji svi setuju posebne flegove

Registar UCBxIV sadrži informaciju o aktivnom zahtevu za prekid najvišeg nivoa

### RX, TX i prekidi promene stanja

Interrupt Flag	Interrupt Condition
UCALIFG	Arbitration-lost. Arbitration can be lost when two or more transmitters start a transmission simultaneously, or when the USCI operates as master but is addressed as a slave by another master in the system. The UCALIFG flag is set when arbitration is lost. When UCALIFG is set, the UCMST bit is cleared and the I <sup>2</sup> C controller becomes a slave.
UCNACKIFG	Not-acknowledge interrupt. This flag is set when an acknowledge is expected but is not received. UCNACKIFG is automatically cleared when a START condition is received.
UCSTTIFG	START condition detected interrupt. This flag is set when the I <sup>2</sup> C module detects a START condition together with its own address while in slave mode. UCSTTIFG is used in slave mode only and is automatically cleared when a STOP condition is received.
UCSTPIFG	STOP condition detected interrupt. This flag is set when the I <sup>2</sup> C module detects a STOP condition while in slave mode. UCSTPIFG is used in slave mode only and is automatically cleared when a START condition is received.

# I<sup>2</sup>C prekidi 2/2

```
USCI_I2C_ISR
    ADD    &UCB0IV, PC    ; Add offset to jump table
    RETI    ; Vector 0: No interrupt
    JMP    ALIFG_ISR      ; Vector 2: ALIFG
    JMP    NACKIFG_ISR    ; Vector 4: NACKIFG
    JMP    STTIFG_ISR     ; Vector 6: STTIFG
    JMP    STPIFG_ISR     ; Vector 8: STPIFG
    JMP    RXIFG_ISR      ; Vector 10: RXIFG
TXIFG_ISR    ; Vector 12
    ...      ; Task starts here
    RETI    ; Return
ALIFG_ISR    ; Vector 2
    ...      ; Task starts here
    RETI    ; Return
NACKIFG_ISR  ; Vector 4
    ...      ; Task starts here
    RETI    ; Return
STTIFG_ISR   ; Vector 6
    ...      ; Task starts here
    RETI    ; Return
STPIFG_ISR   ; Vector 8
    ...      ; Task starts here
    RETI    ; Return
RXIFG_ISR    ; Vector 10
    ...      ; Task starts here
    RETI    ; Return
```

