

SERIJSKI PERIFERNI INTERFEJS (SPI)

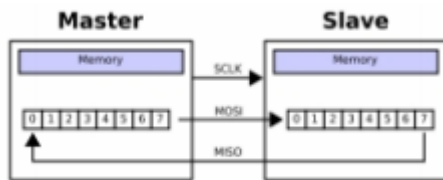
1.OPIS

1.1 PREGLED SPI

SPI je jednostavan način serijskog sinhronog prenosa podataka koji se koristi za komunikaciju na kratkoj udaljenosti, primarno u sistemima za ugradnju(**embedded systems**). Interfejs je nastao u kompaniji Motorola, prvenstveno za spajanje mikrokontrolera sa periferijama (uloga je slična I²C standardu i jednožičnoj vezi). Tipična primena uključuje senzore *Secure Digital* kartice i LCD monitore.

SPI se naziva još i četvorožična veza zbog četiri signala koji učestvuju u komunikaciji. SPI ostvaruje dvosmernu, dupleks vezu između jednog glavnog uređaja (*master*) i jednog ili više sporednih uređaja (*slaves*). SPI se može opisati kao sinhroni serijski interfejs, ali je drugačiji od *Synchronous Serial Interface* (SSI) protokola koji je takođe četvorožični, ali je simpleks i koristi drugačiju signalizaciju.

Glavni, koji je, po pravilu, mikrokontroler, generiše takt (SCLK) koji je između 1MHz i 70MHz. I glavni i sporedni uređaji imaju u sebi osmootbitni (mada su dozvoljene i druge veličine) pomerački registar koji su



povezani na slici levo. Komunikacija počinje kada glavni računar aktivira signal SS(*slave select*), to jest postavi logičku nulu na tu liniju (signal SS nije naznačen na slici). Komunikacija je uvek dvosmerna, jedan bajt ide od glavnog ka sporednom, drugi bajt istovremeni se prenosi od sporednog ka glavnom. Oba prenosa se obavljaju istovremeno čak i kad neki od njih nema smisla. Na

primer, ako mikrokontroler šalje veći broj bajtova ka periferiji, a periferija nema podataka koje bi slala mikro-kontroleru, smer od periferije ka mikrokontroleru prenosi podatke koji nemaju smisla, ali se ipak prenose.

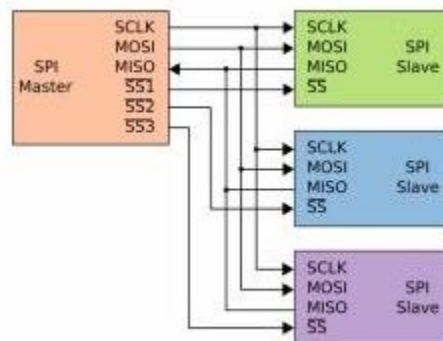
Signali koji učestvuju u komunikaciji su sledeći:

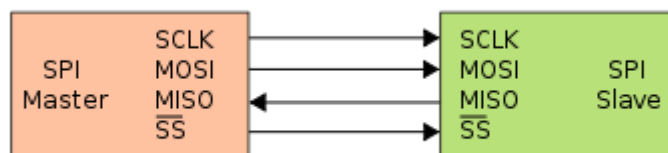
SCLK – Signal takta koji generiše glavni. Smer je od glavnog ka sporednom.

MOSI – *Master Output Slave Input*. Podatak od glavnog ka sporednim uređajima.

MISO – *Master Input Slave Output*. Podatak od sporednog ka glavnom uređaju.

SS– *Slave Select*. Signal generiše glavni. Nula na ovomvodu otpočinje komunikaciju.





Veza sa više sporednih uređaja zahteva da glavni (*master*) obezbedi posebne *SS(slave select)* signale za svaki sporedni (*slave*). Ovakva veza zahteva i da sporedni uređaju na svom izlazu MISO imaju kolo sa stanjem visoke impedanse kako ne bi došlo do sukoba više izlaza koji su povezani za istu tačku.

Postoji mogućnost i povezivanja više sporednih uređaja u lanac (*daisy chain*) kada je izlaz glavnog vezan na ulaz prvog sporednog, njegov izlaz za ulaz drugog sporednog i tako dalje. Izlaz poslednjeg sporednog u lancu vezan je za ulaz glavnog.

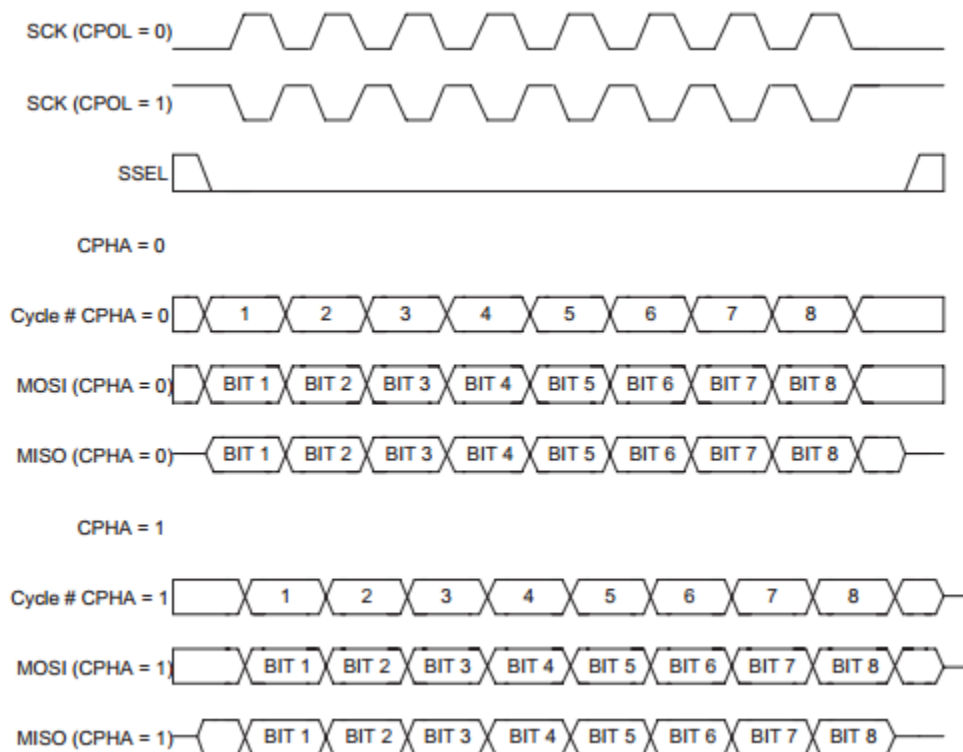
Tehnike komunikacije označene sa SSI, SSP i MicroWire su izvedene iz SPI.

Ukratko:

SPI je serijski interfejs sa potpunim dupleksom. Podržava rad sa više nadređenih i podređenih (u daljem tekstu master i slejv) učesnika na istoj magistrali. Prenos podataka se može obavljati samo između jednog mastera i jednog slejva, pri čemu master uvek šalje od 8 do 16 bita podataka slejvu, dok slejv šalje istu količinu podataka.

1.2 PRENOS PODATAKA

Na sledećoj slici je vremenski dijagram na kom su prikazana četiri različita formata prenosa podataka koje SPI podržava. Prikazan je prenos osmobitnog podatka. Dijagram je podeljen na tri horizontalna dela. Prvi deo opisuje signale SCK i SSEL. Drugi deo opisuje signale MOSI i MISO kada je CPHA 0. Treći deo opisuje MOSI i MISO signale kada je CPHA 1.



Kombinacije CPOL i CPHA bita kontrolnog registra S0SPCR određuju sledeće:

- 1) Kada će biti poslat prvi bit;
- 2) Kada će biti poslani ostali biti;
- 3) Koja ivica takta će na prijemu očitavati stanje bita.

CPOL	CPHA	Prvi podatak poslat	Ostali podaci poslani	Podatak uzorkovan
0	0	Pre prve uzlazne ivice SCK takta	Opadajuća ivica SCK takta	Uzlazna ivica SCK takta
0	1	Prva uzlazna ivica SCK takta	Uzlazna ivica SCK takta	Opadajuća ivica SCK takta
1	0	Pre prve opadajuće ivice SCK takta	Uzlazna ivica SCK takta	Opadajuća ivica SCK takta
1	1	Prva opadajuća ivica SCK takta	Opadajuća ivica SCK takta	Uzlazna ivica SCK takta

Početak i kraj osmobitnog prenosa zavisi od toga da li je periferija master ili slejv i koja je vrednost na CPHA.

Ako je periferija master, za početak prenosa dovoljno je da master ima spreman bajt podataka za prenos. Od tog momenta, master može aktivirati takt i otpočeti prenos. Prenos se završava kada se završi poslednji ciklus prenosa podataka.

Ako je periferija slejv i CPHA je postavljen na 0, prenos počinje kada je aktivan signal SSEL, a završava se kada signal SSEL postane neaktivan. Ukoliko je CPHA postavljen na 1, prenos podataka će krenuti na prvu ivicu takta kada je slejv izabran, a završiće se na poslednju ivicu takta.

1.3 OPŠTE INFORMACIJE

Postoje četiri registra koja kontrolišu SPI periferiju.

SPI kontrolni registar sadrži broj programabilnih bitova koji određuju funkciju SPI bloka. Bitno je podesiti registar pre prenosa podataka.

SPI statusni registar sadrži bite (koji se mogu samo čitati) kojima se prati status SPI interfejsa. Osnovna uloga ovog registra je da detektuje kraj prenosa podataka. Konkretno, na ovo ukazuje SPIF bit. Preostali biti služe za indikaciju specijalnih slučajeva.

SPI registar podataka sadrži podatke za prenos. Za serijski prenos podataka koristi se interni pomerački registar. Ne postoji bafer između registra podataka i internog pomeračkog registra. Sadržaj registra podataka prebacuje se direktno u interni pomerački registar. Prema tome, podatke ne treba upisivati kad je prenos u toku. Podaci za čitanje se smeštaju u bafer. Kada se prenos završi, primljeni podaci se prebacuju u jedno-bajtni bafer podataka, iz kojeg se kasnije čitaju.

SPI registar brojača takta kontroliše uč estanost takta kada je SPI blok u master modu. Ova vrednost se mora podesiti pre početka prenosa podataka. Ovaj registar nema nikakvu funkciju ako je SPI blok u slejv modu.

Ulazi/izlazi u ovoj implementaciji SPI- a su standardni CMOS ulazi/izlazi. Ako je uređaj u slejv modu, njegovi ulazi/izlazi će biti aktivni samo ako je njegov SSEL signal aktivan.

1.4 OPERACIJA MASTERA

Sekvenca prenosa podataka na strani mastera:

1. Postaviti željenu uč estanost takta u registar brojača takta.
2. Podesiti SPI kontrolni registar na željene opcije.
3. Upisati podatak za prenos u SPI registar podataka. Ovaj upis pokreće prenos podataka.
4. Sačekati da SPIF bit u SPI statusnom registru bude postavljen na 1.
5. Pročitati SPI statusni registar.
6. Pročitati primljeni podatak iz SPI registra podataka (opciono).
7. Za prenos dodatnih podataka krenuti od tačke 3.

Da bi se SPIF bit postavio na 0, potrebno je pročitati ili upisati podatke u SPI registar podataka.

1.5 OPERACIJA SLEJVA

Sekvenca prenosa podataka na strani slejva:

1. Podesiti SPI kontrolni registar na željene opcije.
2. Upisati podatak za prenos u SPI registar podataka (opciono).
3. Sačekati da SPIF bit u SPI statusnom registru bude postavljen na 1. SPIF bit će se postaviti nakon poslednje ivice takta prenosa podataka.
4. Pročitati SPI statusni registar.
5. Pročitati primljeni podatak iz SPI registra podataka (opciono).
6. Za prenos dodatnih podataka krenuti od tačke 2.

Da bi se SPIF bit postavio na 0, potrebno je pročitati ili upisati podatke u SPI registar podataka.

2. SPECIJALNI SLUČAJEVI

2.1 PRESORO ČITANJE (READ OVERRUN- ROVR)

Ovo se dešava kada interni bafer za čitanje sadrži podatke koje procesor još nije pročitao, a novi podaci pristignu. Ako je u statusnom registru SPIF bit aktivan podaci u baferu za čitanje su validni. Kada se prenos završi, SPI blok treba da prebaci primljeni podatak u bafer za čitanje. Ako je SPIF bit aktivan (bafer za čitanje je pun), novopristigli podatak biće izgubljen, a ROVR bit će biti aktivan u statusnom registru.

2.2 PREBRZO UPISIVANJE (WRITE COLLISION-WCOL)

Kao što je već rečeno, podaci se ne smeju pisati u registar podataka dok je prenos u toku, jer ne postoji bafer između registra podataka i internog pomeračkog registra. U suprotnom će podaci biti izgubljeni (write collision) i WCOL bit će biti aktivan u statusnom registru.

2.3 GREŠKA U MODU (MODE FAULT-MODF)

Kada je SPI blok u master modu, signal SSEL mora biti neaktivan. Ukoliko bi u toj situaciji signal SSEL bio aktivan, to bi značilo da je neki drugi master izabrao ovaj SPI blok za slejv. Ova situacija se naziva greška u modu. Kad nastupi greška u modu, u statusnom registru će se aktivirati MODF bit, SPI signal dražveri se isključuju i SPI blok prelazi u slejv mod.

2.4 PREKID RADA SLEJVA (SLAVE ABORT-ABRT)

Ako signal SSEL pređe u neaktivno stanje pre nego što se prenos završi, nastupa prekid rada slejva. U tom slučaju dolazi do gubitka poslatih i primljenih podataka i u statusnom registru će se aktivirati ABRT bit.

3. OPIS PINOVA

	MASTER	SLAVE
SCK0	OUT	IN
SSEL0	/	IN
MISO0	IN	OUT
MOSI0	OUT	IN

SSEL0 ukazuje na slejv koji će učestvovati u razmeni podataka. Svaki slejv ima jedinstven ulaz. Stanje „/” označava da je **SSEL0** na strani mastera uvek neaktivan. Master ne prima **SSEL0** signal, master ga generiše.

SCK0 služi za sinhronizaciju prenosa podataka putem SPI-a.

Baferovanje prilikom upisa podataka ne postoji. Podatak se upisuje direktno u šift registar i “gazi” prethodni(*write collision*).

Baferovanje kod čitanja podataka postoji.

Pin	Tip	Opis
SCK0	Ulaz/izlaz	Serial Clock. Serijski takt. Služi za sinhronizaciju prenosa podataka preko SPI-a. SPI uvek pokreće master, a prijemnik je slejv. Može se programirati da radi na visokom ili niskom nivou. SPI je jedino aktivan prilikom prenosa podataka. U bilo kojem drugom trenutku, može biti ili neaktivan ili u stanju visoke impedanse.
SSEL0	Ulaz	Slave Select. Odabir slejva. Ovaj signal je aktivan na niskom nivou i ukazuje koji će slejv učestvovati u razmeni podataka. Svaki slejv ima jedinstven SSEL ulaz. SSEL mora biti na niskom nivou pre početka razmene podataka i tako treba da ostane i za vreme razmene podataka. Ako u toku prenosa signal SSEL pređe u visoki nivo, prenos će se prekinuti. U ovom slučaju slejv odbacuje sve podatke i čeka novi početak. Osim što je slejv prešao u stanje čekanja, ne postoji nikakva druga indikacija za ovaj slučaj. Na strani mastera mora da bude neaktivan sve vreme. Master ne pokreće SPI direktno, nego se to obično softversi radi preko GPIO.
MISO0	Ulaz/izlaz	Master In Slave Out. Master prima slejv šalje. Jednosmerni signal koji se koristi za serijski prenos podataka od slejva ka masteru. Ako je uređaj slejv, serijski podatak je izlaz na ovom pinu, a ako je master, serijski podatak je ulaz na ovom pinu. Kada nije određen slejv, ovaj pin se nalazi u stanju visoke impedanse.
MOSI0	Ulaz/izlaz	Master Out Slave In. Master šalje slejv prima. Jednosmerni signal koji se koristi za serijski prenos podataka od mastera ka slejvu. Ako je uređaj master, serijski podatak je izlaz na ovom pinu, a ako je slejv, serijski podatak je ulaz na ovom pinu.

4. OPIS REGISTRARA

SPI sadrži pet registara prikazanih u sledećoj tabeli.

Naziv	Opis	Pristup	Vrednos posle reseta
S0SPCR	SPI Control Register. SPI kontrolni registar. Ovaj registar kontroliše rad SPI-a.	R/W	0x00
S0SPSR	SPI Status Register. SPI statusni registar. Ovaj registar pokazuje stanje SPI-a.	RO	0x00
S0SPDR	SPI Data Register. SPI registar podataka. Ovaj bidirekcionni registar omogućava slanje i prijem podataka preko SPI-a. Predaja se ostvaruje upisivanjem u ovaj registar, a takođe, primljeni podaci se mogu pročitati iz ovog registra.	R/W	0x00
S0SPCCR	SPI Clock Counter Register. SPI registar brojača takta. Ovaj registar kontroliše učestanost master takta.	R/W	0x00
S0SPINT	SPI Interrupt Flag. SPI registar prekida. Ovaj registar sadrži fleg prekida za SPI.	R/W	0x00

4.1 SPI KONTROLNI REGISTAR (S0SPCR)

Ovaj registar kontroliše rad SPI0 na osnovu konfiguracije bitova.

Bit	Simbol	Vrednost	Opis	Vrednos posle reseta
1:0	---		Rezervisano. Korisnik ne bi smeo da upisuje 1 na mestu ovih bita. Vrednost koja se čita nije definisana.	nedefinisana
2	BitEnable	0	SPI kontroler šalje i prima 8 bita podataka po prenosu.	0
		1	SPI kontroler šalje i prima onoliko bita podataka po prenosu koliko je definisano bitima 11:8.	
3	CPHA	0	Podaci se uzorkuju na prvu ivicu SCK takta. Prenos počinje i završava se aktivacijom i deaktivacijom SSEL signala.	0
		1	Podaci se uzorkuju na drugu ivicu SCK takta. Prenos počinje sa prvom ivicom, a završava se zadnjom ivicom takta kada je SSEL signal aktivan.	
4	CPOL	0	SCK je aktivan u visokom nivou.	0
		1	SCK je aktivan u niskom nivou.	
5	MSTR	0	SPI radi u slejv modu.	0
		1	SPI radi u master modu.	
6	LSBF	0	SPI prenos počinje sa MSB bitom (bit 7).	0
		1	SPI prenos počinje sa LSB bitom (bit 0).	
7	SPIE	0	SPI prekidi su ignorisani.	0
		1	Dešava se hardverski prekid svaki put kada su aktivni SPIF ili MODF biti.	
11:8	BITS		Kada je bit 2 ovog registra 1, ovo polje određuje broj bita za prenos.	0000
		1000	8 bita	
		1001	9 bita	
		1010	10 bita	
		1011	11 bita	
		1100	12 bita	
		1101	13 bita	
		1110	14 bita	
		1111	15 bita	
		0000	16 bita	
15:12	---		Rezervisano. Korisnik ne bi smeo da upisuje 1 na mestu ovih bita. Vrednost koja se čita nije definisana.	nedefinisana

4.2 SPI STATUSNI REGISTAR (S0SPSR)

Bit	Simbol	Opis	Vrednos posle reseta
2:0	---	Rezervisano. Korisnik ne bi smeo da upisuje 1 na mestu ovih bita. Vrednost koja se čita nije definisana.	nedefinisana
3	ABRT	Slave abort. Slejev prekinut. Kada ima vrednost 1, ukazuje da se desilo prekidkomunikacije opisan u poglavlju 2.4. Ovaj bit se briše čitanjem ovog registra.	0
4	MODF	Mode fault. Greška u modu. Kada ima vrednost 1, ukazuje da se desila greška u modu. Ovaj bit se briše čitanjem ovog registra i zatim upisivanjem u SPI0 controlni registar.	0
5	ROVR	Read overrun. Presporo čitanje. Kada ima vrednost 1, ukazuje da se izgubio podatak koji nije pročitao. Ovaj bit se briše čitanjem ovog registra.	0
6	WCOL	Write collision. Prebrz upis. Kada ima vrednost 1, ukazuje da se podatak prepisao pre nego što je poslat. Ovaj bit se briše čitanjem ovog registra i zatim pristupanjem SPI registru podataka.	0
7	SPIF	SPI transfer complete flag. Prenos podataka uspešan. Kada ima vrednost 1, ukazuje da je prenos podataka bio uspešan. Kod mastera, ovaj bit se setuje na kraju poslednjeg takta prenosa podataka. Kod slejva, ovaj bit se setuje na zadnju ivicu (takta) uzorka. Ovaj bit se briše čitanjem ovog registra i zatim pristupanjem SPI registru podataka. Napomena: ovo nije SPI prekidni fleg. On se nalazi u SPINT registru!	0

4.3 SPI REGISTAR PODATAKA (S0SPDR)

Ovaj bidirekcionni registar omogućava slanje i prijem podataka preko SPI-a. Predaja se ostvaruje upisivanjem u ovaj registar, a takođe i primljeni podaci se mogu pročitati iz ovog registra. Ako master upiše podatak u ovaj registar pokrenuće prenos podataka. Upis u ovaj registar će biti blokiran od trenutka kada se prenos pokrene do trenutka kada se SPIF bit setuje a statusni registar ne iščita.

Bit	Simbol	Opis	Vrednos posle reseta
7:0	DataLow	SPI bidirekcionni port podataka.	0x00
15:8	DataHigh	Ako bit 2 u SPCR registru ima vrednost 1, a biti 11:8 su različiti od 1000 (tj. ako je dužina podatka veća od 8 bita), ovi biti sadrže podatke koji se prenose.	0x00

4.4 SPI REGISTAR BROJAČA TAKTA (S0SPCCR)

Ovaj registar kontroliše učestanost master takta. S0SPCCR registar pokazuje broj potrebnih PCLK ciklusa neophodnih za jedan SPI takt. Vrednost ovog registra uvek mora biti paran ceo broj, koji je jednak ili veći od 8. U suprotnom, može doći do nepredvidivog ponašanja.

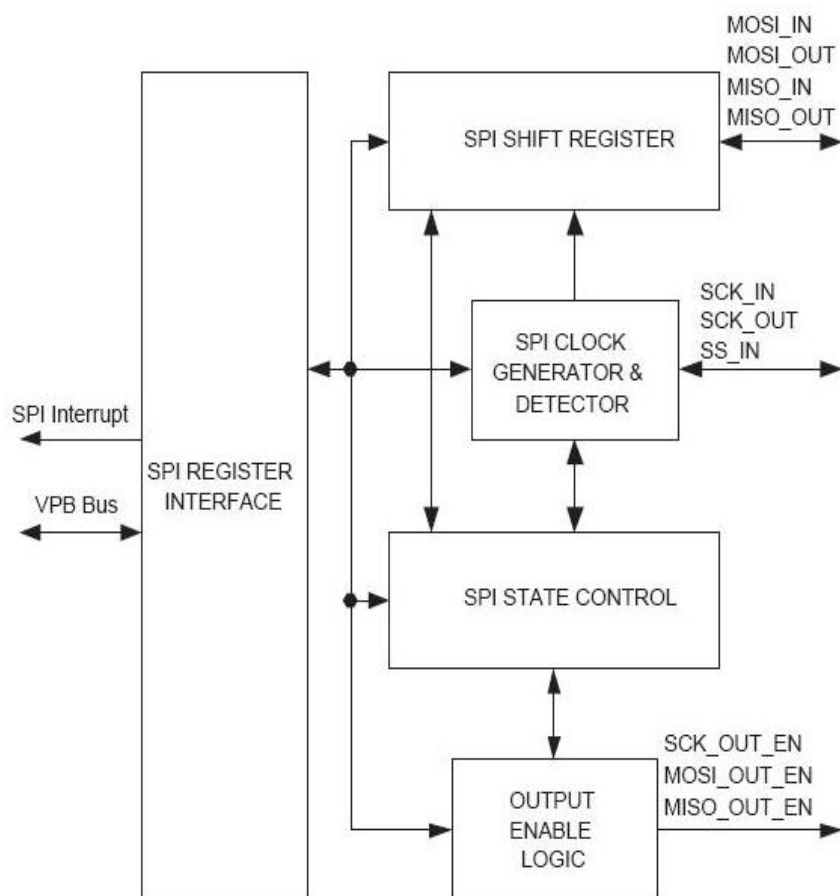
Bit	Simbol	Opis	Vrednos posle reseta
7:0	Counter	Vrednost sa kojom se deli PCLK da se dobije takt SPI0.	0x00

4.5 SPI REGISTAR PREKIDA (S0SPINT)

Ovaj registar sadrži fleg prekida za SPI0.

Bit	Simbol	Opis	Vrednos posle reseta
0	SPI Interrupt Flag	SPI fleg prekida. Ovaj bit postavlja SPI kada zahteva prekid. Briše se upisivanjem 1 u ovaj bit. Napomena: Ovaj bit će se jednom postaviti kada je SPIE = 1 i barem jedan od SPIF i WCOL bita ima vrednost 1. Međutim, samo kada je ovaj bit postavljen i SPI0 prekid uključen u VIC registru, softver za obradu prekida će obraditi SPI prekid.	0x00
7:1	---	Rezervisano. Korisnik ne bi smeo da upisuje 1 na mestu ovih bita. Vrednost koja se čita nije definisana.	nedefinisana

5. BLOK ŠEMA



6. Primer

U sledećem primeru prikazano je kako se posredstvom SPI periferije upravlja radom D/A konvertora. Program će da omogući generisanje signala rampe na izlazu D/A konvertora. Rampa se aproksimira sa 32 odbirka a njena učestanost treba da bude 10Hz. Program na izlazu D/A konvertora generiše rampu ako se preko UART0 primi karakter '1' a ako se primi bilo koji drugi karakter na izlazu D/A postavlja se VCC/2. Primljeni karatkter se vraća.

#include "user.h"
#include "lpc2148.h"
#include "string.h"
#include "stdio.h"

Za potrebe programa, potrebno je pored header datoteke za lpc2148 uključiti i standardnu C biblioteku za ulaz i izlaz, kao i biblioteku za operacije sa stringovima.

void Init_GPIO(void);
void Start_TMR0(void);
void Init_PCB(void);
void Init_UART0(void);
void Init_SSP(void);

Potrebno je inicijalizovati Tajmer, PCB, GPIO, UART I SSP periferiju.

#define SEK (U32)(14745600/4)
#define DELTAT (U32)(SEK/10/32)
#define DELTAV (U32)(0xFFFF/32)
#define VCCPOLA (U32)(0xFFFF/2)
U8 Rx;
U32 brojac;
U16 ADD_vrednost;

Takođe je izvršena definicija konstanti i deklaracija promenljivih.

void main(void)
{
Init_PCB();
Init_GPIO();
Init_UART0();
Init_SSP();
Start_TMR0();

Pre svega, u glavnoj funkciji potrebno je pozvati sve funkcije za inicijalizaciju periferija.

U32 var=0x0;
char Status;
while(1)
{
if(UOLSR&0x1)
{
Rx=UORBR;
UOTHR=Rx;
if(Rx!='1') SSPDR=0x7FFF;
}
if((TOTC>DELTAT)&&(Rx=='1'))
{
Start_TMR0();
Status = (char) SSPSR;
SSPDR=var;
var+=DELTAV;
}
}
}

Ukoliko se preko UART0 primi '1' generiše se “rampa”, a ako se primi bilo koji drugi karatker na izlazu D/A je VCC/2

void Init_GPIO(void)
{
IODIRO = 1<<4;
IODIRO = 1<<5;
IODIRO = 1<<6;
IODIRO = 1<<7;
}
void Start_TMR0(void)
{
TOTCR=3;
TOTCR=1;
}
void Init_SSP(void)
{
SSPCPSR=0xA;
SSPCR0=0xCF;
SSPCR1=0x2;
}

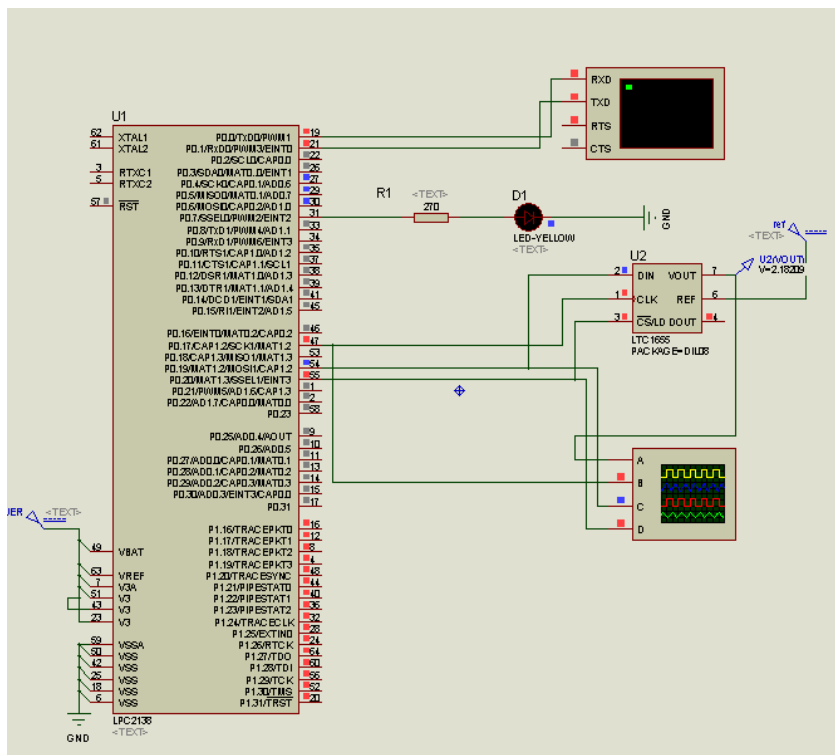
U definiciji funkcije Init_SSP podešen je režim rada (master, slave), bitska brzina i broj informacionih bitova.

void Init_PCB(void)
{
PINSEL0 = 0x5; //Tx0 Rx0 definisane kao izlazne linije
PINSEL1 = (0x1<<3); //SCK1 (SSP)
PINSEL1 = (0x1<<7); //MOSI1 (SSP)
PINSEL1 = (0x1<<9); //SSEL1 (SSP)
}

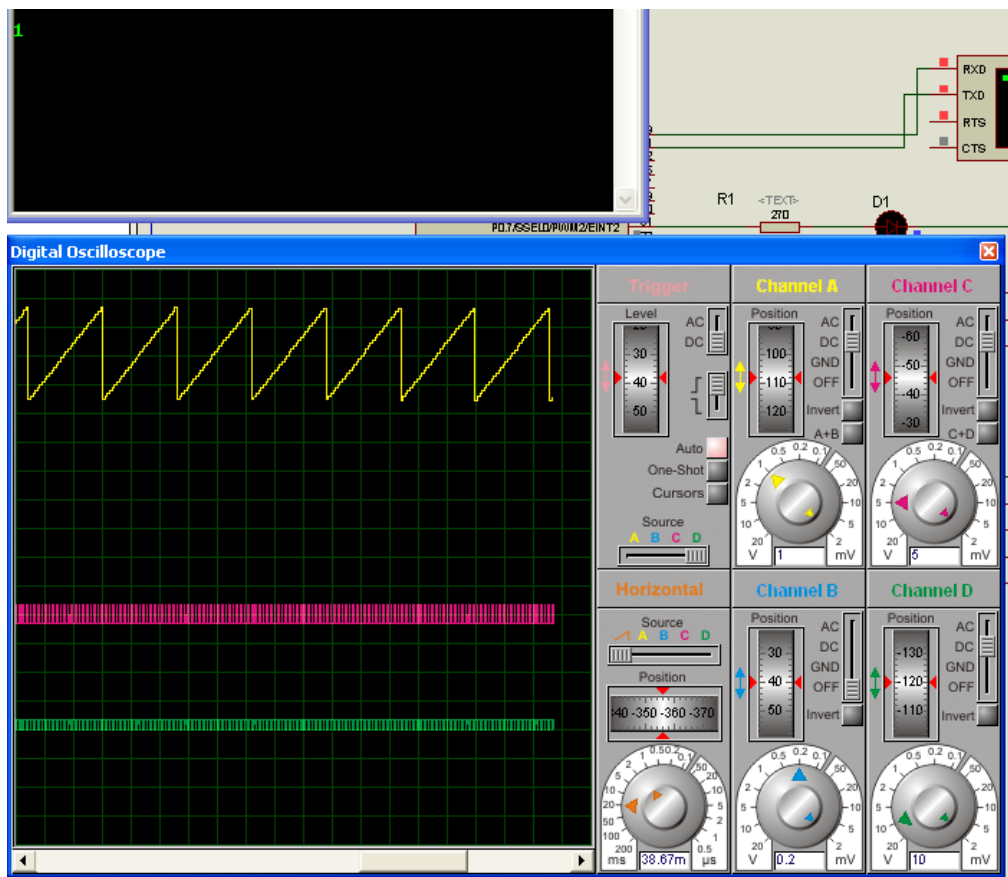
Po uključenju napajanja, sve I/O nožice imaju funkciju GPIO periferije. Ukoliko želimo da ih koristimo kao ulaze/izlaze nekih drugih periferija mikrokontrolera, potrebno je inicijalizovati *Pin Connect Block*, setovanjem odgovarajućih pinova registra PINSEL0 i PINSEL1.

void Init_UART0(void)
{
U0IER = 0;
U0LCR = 0x3;
U0LCR = 0x80;
U0DLM = 0x0;
U0DLL = 0x18;
U0LCR &= 0xFFFFF7F;
}

Kod funkcije za inicijalizaciju UART-a, obrisan je interapt, broj informacionih karaktera postavljen na 8, bodska brzina na 9600 i na kraju je zatvoren pristup Divisor registru.



Slika: Simulator



Slika: Izlaz na digitalnom osciloskopu.