

Za osnovu sistema automatizacije procesa proizvodnje predviđen je jedan centralni industrijski računar, master, realizovan na bazi standardnog SoCa. U procesu proizvodnje su identifikovana različita radna mesta koja se mogu podeliti u više kategorija, ali je ustanovljeno da je za njih potrebno hardverski četiri različite grupe uređaja. Shodno tome, radna mesta su podeljena u četiri grupe sa maksimalno 10 uređaja po grupi. Sa računarom komuniciraju preko zajedničke standardne **pundupleks** asinhronne serijske magistrale.

SoC koji je na raspolaganju za realizaciju delova sistema je bez spoljne magistrale, sa četiri 8-bitna paralelna bidirekciona portova. Unutar SoCa se nalazi programabilni UART koji može da se programira da radi sa 7, 8 ili 9bitnim porukama, sa automatskim prepoznavanjem adrese, sa bitom parnosti, ... i van SoCa izlazi preko namenskih pinova Tx i Rx. Paralelni portovi mogu da se konfigurišu da rade kao ulazni prekidni zahtevi aktivni sa ivicom ili nivoom.

Svaka grupa uređaja je korišćenjem paralelnih portova ulančana. I svaka grupa uređaja pored linija za serijsku komunikaciju ima još jednu dodatnu liniju koja ide ka centralnom računaru, preko koga uređaji prekidom obaveštavaju centralni računar o incidentnim situacijama. U svakom uređaju se nalazi 64 8bitne interne memorijske lokacije koje je moguće čitati i u njih upisati neke podatke.



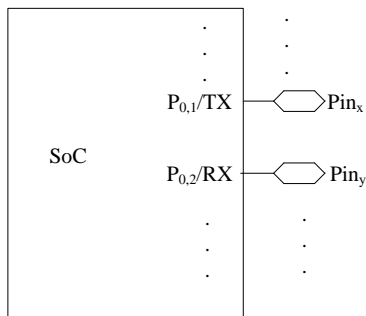
1

1. [5] Nacrtati dodatan hardver koji obezbeđuje ulančavanje i izlazak na zajedničku liniju za prekid. Kako se konfigurišu portovi mastera, na koje dolaze te zajedničke linije. Kako obezbediti mogućnost da programi za SoC unutar iste grupe uređaja budu identični? Očigledno je da treba iskoristiti ideju pseudovektorskih prekida, gde svaki OU (koji ima aktivan prekidni zahtev i najvećeg je prioriteta) odgovara svojim kodom. Ako se taj kod "hardkoduje" unutar programa onda svaki OU ima svoj program. Cilj je da imaju isti program.

2. [15] Definisati formate i protokole rada na serijskoj magistrali, tako da se obezbedi što pouzdaniji upis i čitanje u svim uređajima. Takođe definisati protokol i formate podataka ako svaka grupa uređaja može o incidentnim situacijama da preko prekidnog zahteva informiše računar. Potrebno je pouzdano identifikovati uređaj(e) koji je poslao prekidni zahtev. Opisati programske zahteve za rad sa tim prekidima unutar uređaja, kao i unutar centralnog računara.



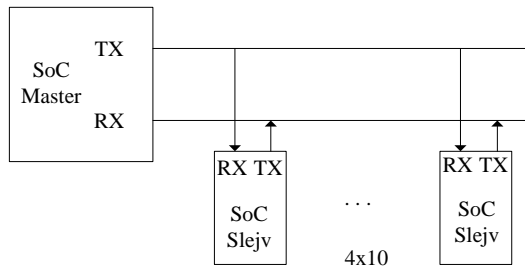
2



Kao što smo rekli, jako česta situacija u integriranim komponentama da bi se uštedelo na fabrikaciji - ceni. Multipleksirani pinovi – manji broj pinova. Pinovi koji mogu da „koriste“ različite unutrašnje periferije. Postoji „negde“ (na odgovarajućoj memorijskoj lokaciji) konfiguracioni registar kojim se definiše koja periferija „izlazi, koristi“ te zajedničke pinove. Obično po resetu to jeste paralelni port. I da ne zaboravimo da postoji i konfiguracioni registar da li je taj bidirekcionni port ulazni i izlazni. Ako je na primer konfigurisano da pinove koristi UART onda je odmah TX izlazni a RX ulazni. Ne treba podešavanje u konfiguracionom registru paralelnog porta.



3



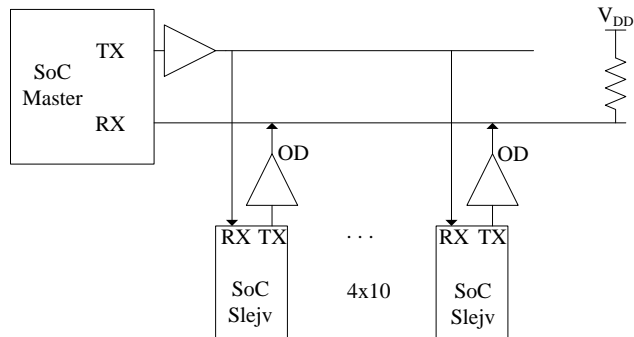
Idejna blok šema

Prva stvar koja se uočava je da su TX linije slejvova (izlazi logičkih kola) povezani na zajedničku liniju – **ŠTO NE SME**
Druga, manje bitna stvar je da je TX izlaz mastera opterećen sa prevelikim brojem ulaza.



4

Razrešenje 1. način

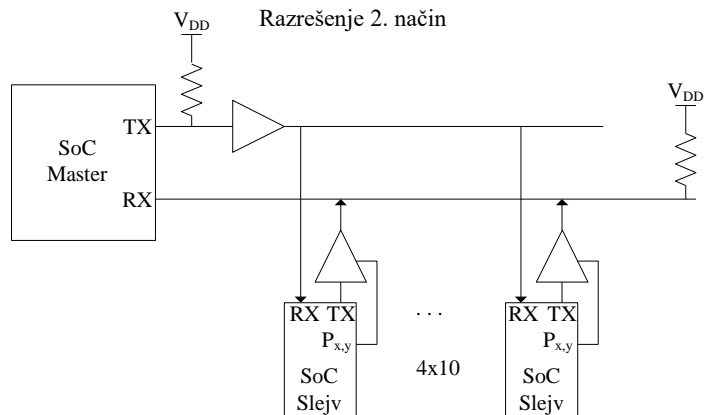


OD – kola sa otvorenim drejnom. Pulup otpornik na zajedničkoj liniji da bi se definisao napon logičke 1. Neaktivno stanje asinhronne serijske komunikacije logička 1. Start bit logička nula. Sjajno za ovo rešenje sa kolima sa otvorenim drejnom. Uočiti da tek „prozvani“ slejv će slati nešto na liniju. I samo on. Da li treba neka druga terminacija zbog prelaznih procesa u vreme reseta. Možda i pulup otpornik na TX liniji mastera, pre bafera koji je stavljen da bi se povećali strujni kapaciteti.



5

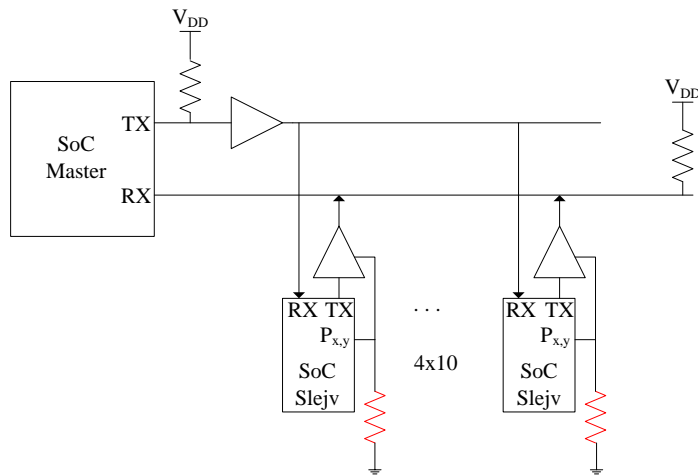
Razrešenje 2. način



Umesto kola sa otvorenim drejnom – trostatička kola. Svaki slejv korišćenjem signala na jednom bitu paralelnog porta $P_{x,y}$ drži trostatičke bafere zatvorenim. Slejv samo kada šalje podatke otvara svoje trostatičke bafere. Znači prvo otvori trostatičke bafere, inicira slanje, sačekava završetak slanja i zatvara trostatičke bafere. Da li ovda fale još neki otpornici?



6

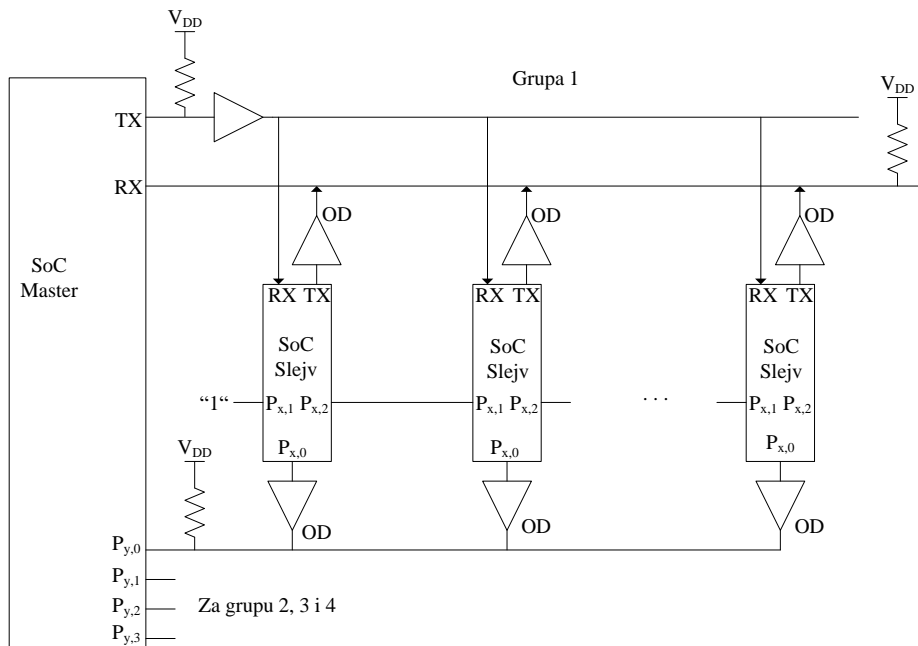


Crveni puldaun otpornici – MORA

Da se podetimo - po resetu ulazni port. Tek u inicijalizaciji upis nule u port - pa izlazni.
 Do tada će se možda zbog smetnji, bez otpornika, otvoriti više trostatičkih bafera na zajedničku liniju.



Da razrešimo prekide



Master

1. Portove $P_{y,0}$ do $P_{y,3}$ u toku inicijalizacije definiše kao ulazne portove sa aktivnim prekidom sa nivoom logičke nule
2. Kada dobije aktivan prekid u okviru prekidne rutine identifikuje koji slejv od 10 iz te grupe je najvišeg prioriteta i ima aktivan prekidni zahtev

Slejv

1. Port $P_{x,0}$ u toku inicijalizacije definiše kao izlazni. Pre toga je u njega upisao logičku 1.
2. Kada ima aktivan prekidni zahtev, koji nije maskiran, postavlja logičku 0 na njega, kao znak masteru.
3. Port $P_{x,1}$ proglašava ulaznim a port $P_{x,2}$ izlaznim i na njega stavlja logičku 1.
4. Kada ima aktivan nemaskiran prekidni zahtev postavlja ga na port $P_{x,0}$ sa aktivnom logičkom nulom.
5. Proverava stanje porta $P_{x,1}$ i ako je ono 0 postavlja 0 i na port $P_{x,2}$. Neko višeg prioriteta ima aktivan zahtev.
6. Ako ima aktivan zahtev postavlja nulu na port $P_{x,2}$.
7. Kada dođe zahtev za identifikaciju od mastera, slejv proverava da li ima aktivan prekidni zahtev i da li mu je dozvoljeno javljanje, najvišeg je prioriteta sa aktivnim zahtevom, $P_{x,1} = 1$. Tada šalje svoju identifikaciju.



Da razrešimo „hardkodovanje“

Svaki slejv mora da ima svoju jedinstvenu adresu da bi se obezbedila komunikacija samo sa njim. Znači u nekakvoj poruci od mastera mora da postoji adresa sa značenjem šaljem samo tebi podatke ili samo od tebe očekujem podatke. Slično je i sa identifikacijom prekida. Svaki slejv ima drugačiju identifikaciju.

Znači negde u programu slejva prilikom prijema pouka mora da stoji:

```
if (primljena_poruka = moja_adresa)
    then ...
    else ignoriši;
```

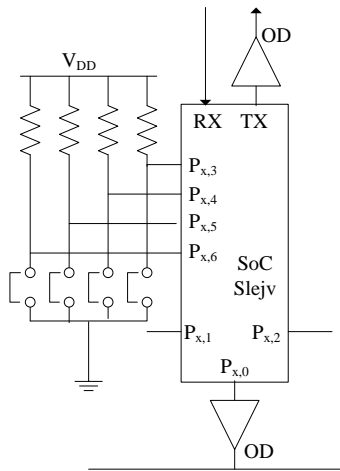
i ovaj podatak moja_adresa je za svakog slejva različit. Znači različiti programi pri čemu prilikom fizičkog programiranja komponente moramo biti oprezni.

Isto važi i za identifikaciju prekida

```
if (ima aktivan prekidni zahtev AND najvišeg je prioriteta)
    then send(moja_adresa);
    else ignoriši;
```

Programi slejvova su identični osim ovog parametra. Konstanta u programu. I zbog njega su različiti. Kako napraviti zaista identične programe?





Slejev u toku inicijalizacije proglašava portove $P_{x,3}$ do $P_{x,6}$ ulaznim i čita četvorobitni podatak. I to je sada **promjenjiva** moja_adresa. Programi slejvova su isti a na projektantu sistema je da prilikom instalacije sistema obezbedi da se pažljivo dodele pojedinačne adresa. Uputstvo!

I sa ovim bi idejni projekat hardvera u ovom delu razrešili. Ostali su protokoli i formati podataka u komunikaciji.



Protokoli i formati podataka.

SoC koji je na raspolaganju za realizaciju delova sistema je bez spoljne magistrale, sa četiri 8-bitna paralelna bidirekciona portova. Unutar SoCa se nalazi programabilni UART koji može da se programira da radi sa 7, 8 ili 9bitnim porukama, sa automatskim prepoznavanjem adrese, sa bitom parnosti, ...

U procesu proizvodnje su identifikovana različita radna mesta koja se mogu podeliti u više kategorija, ali je ustanovljeno da je za njih potrebno hardverski četiri različite grupe uređaja. Shodno tome, radna mesta su podeljena u četiri grupe sa maksimalno 10 uređaja po grupi.

U svakom uređaju se nalazi 64 8bitne interne memorijske lokacije koje je moguće čitati i u njima upisati neke podatke.

A prilikom relazacije hardvera smo uočili potrebu da se maskiraju i demaskiraju prekidni zahtevi, kao i moćnost rada sa pseudovektorskim prekidima.

Imitiramo paralelne magistrale. Mali broj bita je u poruci. Znači zajedničke za adrese i podatke.



Brzina prenosa

Za ovaj zadatak nije bitna ali ...

Brzinu prenosa definišemo taktnim signalom za pomeračke registre. Moraju biti usklađene na predaji i prijemu.

Postoji neki konfiguracioni registar u brojačima, preskalerima, deliteljima kojim definišemo taktni signal za pomeračke registre.

Standardne brzine za ovu vrstu asinhronog prenosa

In telecommunication and electronics, **baud** (*/bɔːd/*; symbol: **Bd**) is a common unit of measurement of symbol rate, which is one of the components that determine the speed of communication over a data channel.

-300Bd
-600Bd
-1200Bd
-2400Bd
-4800Bd
-9600Bd
-19200Bd
-38400Bd
-...
-115200Bd
-...
-921600Bd



... sa automatskim prepoznavanjem adrese ...

Dosta savremenih UART-a ima ovu osobinu

Format poruke za ovu opciju

D8	D7	D6	D5	D4	D3	D2	D1	D0
M	A7	A6	A5	A4	A3	A2	A1	A0

M – Marker bit
M=1 adresa
M=0 nije adresa

UART koji je programiran, konfigurisan, za ovu opciju:

1. U nekom konfiguracionom registru je upisana njegova adresa A7 .. A0
2. Programiran je sa AAR. Prekid od RX dela UARTa je dozvoljen.
3. UART šalje prekid samo ako je primljena poruka sa M=1 i primljena adresa se poklapa sa onom u konfiguracionom registru.
4. Slejv tada zna da je adresiran i može da promeni konfiguraciju tako da prima svaku narednu poruku. Za njega je.
5. Znači „ne uznemiravaju“ ga poruke za druge slejvove.
6. Slabija varijanta, nema konfiguracioni registar za sopstvenu adresu. Uznemiravaju ga samo adresne poruke kada proveri svoju adresu programski.
7. Još bolja varijanta, više ovakvih registara za više adresa, što nama možda i treba.



Počecemo sa 9-bitnim prenosom

Adresiranje

D8	D7	D6	D5	D4	D3	D2	D1	D0
1	G1	G0	A3	A2	A1	A0	-	-

G1 G0 – adresa grupe

A3 .. A0 – adresa unutar grupe 0 .. 9

A3 .. A0 = 1 1 1 1 – zajednička adresa – broadcast adresa
za zajedničke akcije, npr. identifikaciju prekida

D1 D0 za sada neka ostanu neiskorišćeni

kasnije ćemo možda reći da se ignorišu na prijemu ako ih zaista ne iskoristimo

A možda da pomerimo za jedan bit

D8	D7	D6	D5	D4	D3	D2	D1	D0
1	GB	G1	G0	A3	A2	A1	A0	-

GB – broadcast adresa za sve grupe – ima smisla razmišljati i o tome



Puno varijanti – nema prepisivanja Identično rešenje – bilo prepisivanja



Da razrešimo prekide:

1. Master je dobio prekidni zahtev
2. Ulazi u odgovarajuću prekidnu rutinu. Zna od koje je grupe.
3. Šalje poruku, gde je G1 G0 adresa te grupe

D8	D7	D6	D5	D4	D3	D2	D1	D0
1	G1	G0	1	1	1	1	x	x

4. Slejv iz te grupe koji ima aktivan prekidni zahtev i najvišeg prioriteta šalje informaciju

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	G1	G0	A3	A2	A1	A0	x	x

Marker može da bude i 1 i 0. Posle ćemo to videti kada mora da ima odgovarajuću vrednost.

G1 G0 nije neophodno ali što da ne radi pouzdanosti

A3 .. A0 – adresa tog slejva

Na osnovu ovoga master ulazi u deo prekidne rutine povezan sa ovim slejvom.



Da se setimo - Paralelne magistrale – maskiranje na izvorima prekida - Obavezno

I ovde je identično

Da se vratimo unazad i iskoristimo ona nekorisćene bite

D8	D7	D6	D5	D4	D3	D2	D1	D0
1	G1	G0	A3	A2	A1	A0	IE	-

IE = 0 – adresiran slejv maskira svoj prekid

IE = 1 – adresiran slejv dozvoljava svoj prekid

GREŠKA

Ako iz bilo kojeg drugog razloga adresiramo slejv, a biće ih, moramo da vodimo računa o stanju tog bita. Popravka:

D8	D7	D6	D5	D4	D3	D2	D1	D0
1	G1	G0	A3	A2	A1	A0	IE	SIE

SIE = 1 – hoćemo da postavljamo IE

SIE = 0 – ignoriši IE

Za sada ovako, možda ćemo se vratiti nazad ako nam ovi biti budu zanimljiviji za nešto drugo. Uočiti da na ovaj način možemo maskirati sve prekide u grupi.



Adresiranje memorijskih lokacija - 64 8bitne

Posle adresne poruke, master šalje

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	W/R	?	A5	A4	A3	A2	A1	A0

W/R = 1 upis podatka u memorijsku lokaciju sa adresom A5 .. A0

W/R = 0 čitanje podatka sa adrese A5 .. A0

Ako je u pitanju upis podatka posle ove poruke master šalje podatak koji se upisuje

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	D7	D6	D5	D4	D3	D2	D1	D0

Ako je u pitanju čitanje podatka posle ove poruke sljev šalje podatak koji se čita

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	D7	D6	D5	D4	D3	D2	D1	D0



Možda ovaj upitnik iskoristiti za maskiranje, onda bi možda i mogli da čitamo trenutno stanje maske

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	W/R	0	A5	A4	A3	A2	A1	A0

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	W/R	1	IE	x	x	x	x	x

Oslobodili smo ona dva bita u prvoj adresnoj poruci za nešto drugo, ali smo možda izgubili mogućnost da maskiramo sve uređaje u grupi.

Uočiti da grupnim adresiranjem smemo da šaljemo podatke celoj grupi.

I upisujemo podatke u iste memorijske lokacije celoj grupi.

Pa to možemo iskoristiti i za maskiranje cele grupe.

A onda može i

D8	D7	D6	D5	D4	D3	D2	D1	D0
1	GB	G1	G0	A3	A2	A1	A0	-

GRUPNO ČITANJE – SAMO AKO JE ULANČAVANJE – NE MOGU SVI ISTOVREMENO DA ODGOVORE



Pouzdanost prenosa

Nemamo mesta za bit parnosti ili neki drugi zaštitni kod u samim porukama.

Uvešćemo echo protokol i timeout logiku.

Može i ACK + timeout logika.

Prilikom slanja adresne poruke master aktivira timeout logiku.

Ako niko ne odgovori i odradi timeout master proba sa novim adresiranjem.

Ako je identifikacija prekida, to smo već pričali. Samo smo još aktivirali timeout.

Da li ima potrebe da razmišljamo šta ako su u slejvovima nestali prekidni zahtevi kada je master ušao u prekidnu rutinu? Kako beše kod kontrolera prekida. Tada svojom adresom odgovara slejv sa najnižim prioritetom. Ali postoji mogućnost čitanja statusa. A to nismo obezbedili. Da se vratimo:

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	W/R	1	IE	AI	x	x	x	x

Prilikom čitanja informacija o masci ali i o statusu aktivnog prekidnog zahteva.

Ne može baš sve odmah kako treba. U iteracijama i popravkom.



Ako nije identifikacija prekida ili grupno adresiranje adresirani slejv odgovara eho porukom.

Master šalje sledeću poruku ako je sve u redu.

Ako je upis – adresirani slejv odgovara eho porukom i na adresu i na podatak koji master šalje posle uspešno primljene eho poruke.

Ako je čitanje – adresirani slejv odgovara podatkom.

Ovo je samo jedna moguća varijanta. Niti je najbolja niti je jedina.
Ona je samo pravolinijska na osnovu prethodnih znanja.



Kako napraviti sledeći ispitni rok?

Za osnovu sistema automatizacije procesa proizvodnje predviđen je jedan centralni industrijski računar, master, realizovan na bazi standardnog SoCa. U procesu proizvodnje su identifikovana različita radna mesta koja se mogu podeliti u više kategorija, ali je ustanovljeno da je za njih potrebno hardverski četiri različite grupe uređaja. Shodno tome, radna mesta su podeljena u četiri grupe sa maksimalno 10 uređaja po grupi. Sa računarom komuniciraju preko zajedničke standardne **poludupleks** asinhronne serijske magistrale.

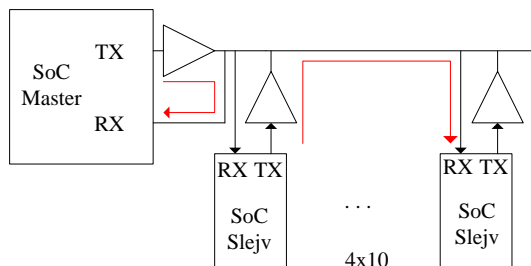
SoC koji je na raspolaganju za realizaciju delova sistema je bez spoljne magistrale, sa četiri 8-bitna paralelna bidirekciona portova. Unutar SoCa se nalazi programabilni UART koji može da se programira da radi sa 7, 8 ili 9bitnim porukama, sa automatskim prepoznavanjem adrese, sa bitom parnosti, ... i van SoCa izlazi preko namenskih pinova Tx i Rx. Paralelni portovi mogu da se konfigurišu da rade kao ulazni prekidini zahtevi aktivni sa ivicom ili nivoom.

Svaka grupa uređaja je korišćenjem paralelnih portova ulančana. I svaka grupa uređaja pored linija za serijsku komunikaciju ima još jednu dodatnu liniju koja ide ka centralnom računaru, preko koga uređaji prekidom obaveštavaju centralni računar o incidentnim situacijama. U svakom uređaju se nalazi 64 **16** bitne interne memorijske lokacije koje je moguće čitati i u njih upisati neke podatke.

Lako



Poludupleks



Što se tiče hardvera nema puno promena.
Kola sa otvorenim drejnom – pulap otpornici itd.
Prekidi i ulančavanje isto.

Uočiti:

1. Poruku koju master šalje slejvovima prima i on.
2. Kada slejv šalje poruku masteru istu poruku prima i on

Ovo možemo da razrešimo troštačim baferima prema RX linijama koji su zatvoreni kada se šalje poruka. Otvaranje i zatvaranje preko bita paralelnog porta. Lako. Ali isto tako je lako razrešiti i softverski. Pruka primljena neposredno posle slanja – ignosrisati. A možda i uporediti pa videti i smetnje na liniji. Ne znači da treba ukinuti eho.

Sada smo se setili. Možda kada je ciklus čitanja bi bilo dobro da i master šalje eho slejvovima, pa ako nešto nije u redu da slejv prekidom signalizira. Tada bi u statusnom registru dodali još bita koji kažu šta je aktiviralo prekid. Ideja za naredni rok. Više izvora prekida u jednom slejvu, a jedan a masteru.

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	W/R	1	IE2	IE1	IE0	AI2	AI1	AI0

Maskiranje svakog
ponaosob. Imamo mesta.

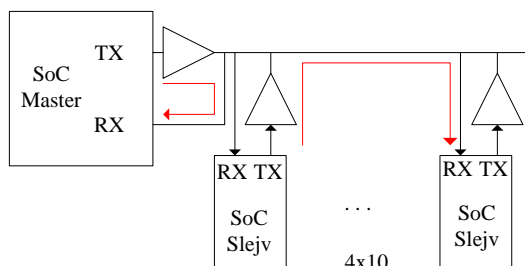


Katedra za elektroniku
prof dr Lazar Saranovac

Namenski računarski sistemi - 2021/22

23

23



Najbitnija stvar: Kada jedan slejv šalje poruku masteru tu poruku primaju i ostali slejvovi!
Ne smeju da je protumače kao adresnu. Česta greška u vašim formatima podataka.

U povratnim porukama od slejva ka masteru pa i u eho porukama mora M=0 kod poludupleks magistrale.

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	D7	D6	D5	D4	D3	D2	D1	D0



Katedra za elektroniku
prof dr Lazar Saranovac

Namenski računarski sistemi - 2021/22

24

24

64 16bitne lokacije

Posle lokalne adresne poruke idu dve poruke sa podacima.
I slejv mora da aktivira timeout logike između poruka.
Upisuje tek kada primi oba dela.

Protokol definiše redolosed poruka.
Na primer prva poruka je viši deo podatka.
Druga poruka je niži deo podatka.

Zašto moruju eho protokol timeout logike itd...

U slučaju gubitka neke poruke kako slejv da razlikuje

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	W/R	0	A5	A4	A3	A2	A1	A0

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	D7	D6	D5	D4	D3	D2	D1	D0



Mogli bi da idemo i na tu stranu ali nam treba mnogo više poruka

Globalno adresiranje

D8	D7	D6	D5	D4	D3	D2	D1	D0
1	1	GB	G1	G0	A3	A2	A1	A0

Lokalno adresiranje – prvi deo adrese pa sada maske

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	1	1						

Lokalno adresiranje – drugi deo adrese ili još više bita ako je adresa velika pa cepkamo

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	1	0						



Prvi deo podatka

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1						

Drugi deo podatka

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0						

Treći deo podatka

...



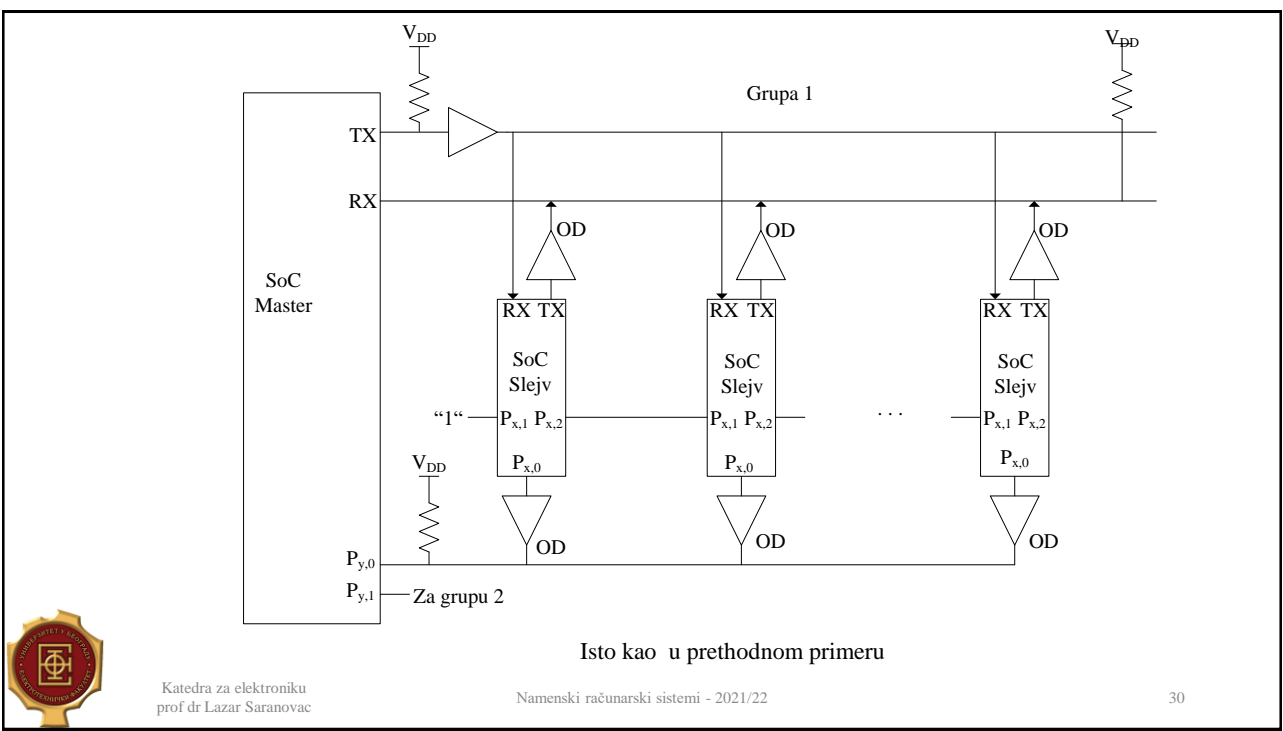
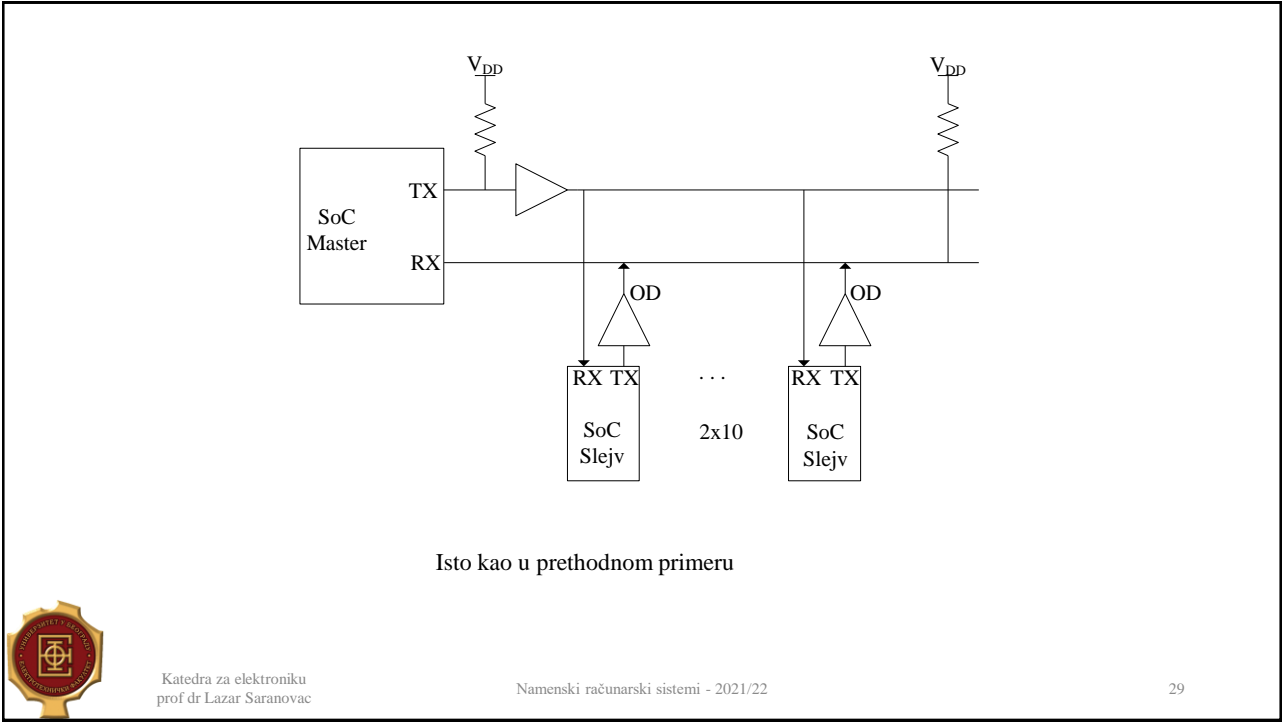
Za osnovu sistema automatizacije nekog procesa proizvodnje predviđen je jedan centralni standardni opštenamenski industrijski računar. U procesu proizvodnje su identifikovana različita radna mesta koja se mogu podeliti u više kategorija, ali je ustanovljeno da je za njih potrebno dve različite grupe uređaja. Shodno tome, radna mesta su podeljena u dve grupe sa maksimalno 10 uređaja po grupi. Sa računarom komuniciraju preko zajedničke standardne pundupleks asinhrona serijske magistrale.

U prvoj grupi se nalaze radna mesta na kojima se na 10cifarskom LED displeju prikazuju numerički podaci dobijeni od strane centralnog računara. Takođe se u okviru ovih radnih mesta nalazi tastatura sa 64 tastera koja je povezana tako da može da se identifikuje pritisak najviše dva tastera. Sva radna mesta iz ove grupe preko zajedničke linije za prekid INT1 obaveštavaju računar ako je detektovan pritisak nekog tastera (prioritet prekida određen je ulančavanjem). Treba omogućiti centralnom računaru da očitava stanje svih eventualno pritisnutih tastera.

U drugoj grupi se nalaze radna mesta na kojima se sa dva jednosmerni motora, koji mogu da se pokreću u oba smera, pozicionira pokretna platforma u dva pravca po X i Y osi. Rezolucija zadavanja pozicije je 1024 inkrementa od minimalne od maksimalne. Ako se dostigne referentni položaj po bilo kojoj osi, što se detektuje zatvaranjem mikroprekidača, prekidom INT2 treba obavestiti centralni računar o tom događaju (prioritet prekida određen je ulančavanjem) kao i o tome koja je referentna tačka dostignuta. Centralni računar može zahtevati informaciju o trenutnom položaju platforme.

Opštenamenski mikrokontroler koji je na raspolaganju za realizaciju delova sistema je bez spoljne magistrale, sa šest 8-bitnih paralelnih bidirekcionih bitadresibilnih portova. Unutar mikrokontrolera i centralnog računara se nalazi programabilni UART koji može da se programira da radi sa 7, 8 ili 9bitnim porukama, sa jednim ili dva stop bita





Master

1. Portove $P_{y,0}$ do $P_{y,1}$ u toku inicijalizacije definiše kao ulazne portove sa aktivnim prekidom sa nivoom logičke nule
2. Kada dobije aktivan prekid u okviru prekidne rutine identifikuje koji slejv od 10 iz te grupe je najvišeg prioriteta i ima aktivan prekidni zahtev

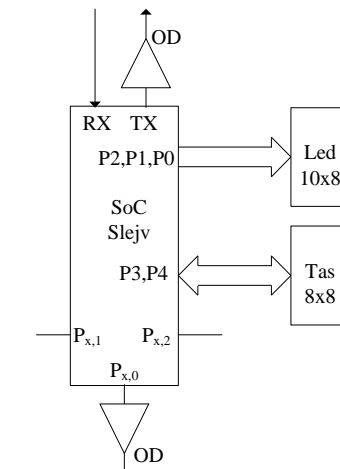
Slejev

1. Port $P_{x,0}$ u toku inicijalizacije definiše kao izlazni. Pre toga je u njega upisao logičku 1.
2. Kada ima aktivan prekidni zahtev, koji nije maskiran, postavlja logičku 0 na njega, kao znak masteru.
3. Port $P_{x,1}$ proglašava ulaznim a port $P_{x,2}$ izlaznim i na njega stavlja logičku 1.
4. Kada ima aktivan nemaskiran prekidni zahtev postavlja ga na port $P_{x,0}$ sa aktivnom logičkom nulom.
5. Proverava stanje porta $P_{x,1}$ i ako je ono 0 postavlja 0 i na port $P_{x,2}$. Neko višeg prioriteta ima aktivan zahtev.
6. Ako ima aktivan zahtev postavlja nulu na port $P_{x,2}$.
7. Kada dođe zahtev za identifikaciju od mastera, slejv proverava da li ima aktivan prekidni zahtev i da li mu je dozvoljeno javljanje, najvišeg je prioriteta sa aktivnim zahtevom, $P_{x,1} = 1$. Tada šalje svoju identifikaciju.



1. Grupa

Displej povezan na standardan način.
Tastaura takođe sa zaštitnim diodama
prema portovima.
Pritisak najviše dva tastera.
Debaunsiranje softverski.



2. Grupa

Jednosmerni motor – upravljanje IŠM.

IŠM generiše mikrokontroler.

Apsolutna pozicija.

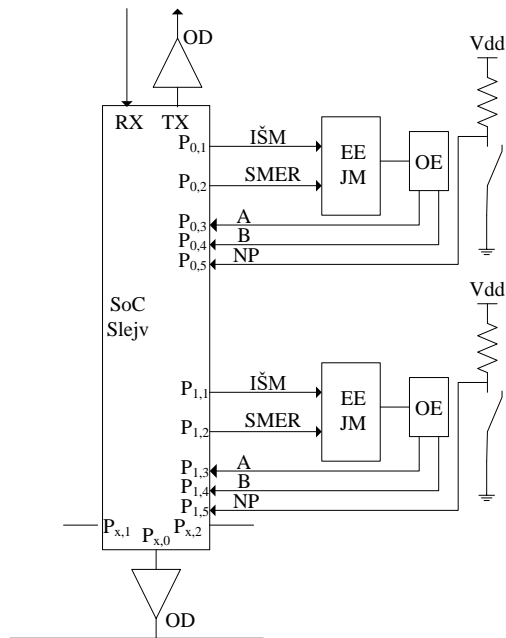
Inkrementalni enkoder sa inicijalizacijom. Dozvoljeno.

Detekciju, debaunsiranje i brojanje radi mikrokontroler.

Meri poziciju i brzinu.

Znači po resetu ili po naredbi se radi inicijalizacija.

Motor se pokrene u jednu stranu, sporo, dok se ne detektuje da je mikroprekidač zatvoren. Tada se proces inicijalizacije završava i čeka naredba od mastera.



Počecemo sa 9-bitnim prenosom

Adresiranje

D8	D7	D6	D5	D4	D3	D2	D1	D0
M=1	G0	A3	A2	A1	A0	-	-	-

G0 – adresa grupe

A3 .. A0 – adresa unutar grupe 0 .. 9

A3 .. A0 = 1 1 1 1 – zajednička adresa – broadcast adresa
za zajedničke akcije, npr. identifikaciju prekida

D2 D1 D0 za sada neka ostanu neiskorišćeni

kasnije ćemo možda reći da se ignorišu na prijemu ako ih zaista ne iskoristimo



Da razrešimo prekide:

1. Master je dobio prekidni zahtev
2. Ulazi u odgovarajuću prekidnu rutinu. Zna od koje je grupe.
3. Šalje poruku, gde je G1 G0 adresa te grupe

D8	D7	D6	D5	D4	D3	D2	D1	D0
1	G0	1	1	1	1	x	x	x

4. Slejv iz te grupe koji ima aktivan prekidni zahtev i najvišeg prioriteta šalje informaciju

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	G0	A3	A2	A1	A0	x	x	x

Na osnovu ovoga master ulazi u deo prekidne rutine povezan sa ovim slejvom.



1. grupa

Posle adresne poruke, master šalje

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	W/R	?	?	?	?	?	?	?

W/R = 1 upis podatka u displej

W/R = 0 čitanje informacije o pritisnutim tasterima

Ako je u pitanju upis podatka u displej šalje adresu displeja

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	1	A3	A2	A1	A0	x	x	x

A nakon nje podatak koji se upisuje – dekodovanje displeja master

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	D7	D6	D5	D4	D3	D2	D1	D0



Ako je u čitanje tastature master šalje

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	x	x	x	x	x	x	x

A slejv odgovara sa dve poruke

Informacija o jednom pritisnutom tasteru

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	x	x	D5	D4	D3	D2	D1	D0

Informacija o drugom pritisnutom tasteru

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	x	x	D5	D4	D3	D2	D1	D0

Ako su u poruci sve 0 nema pritisnutog tastera.

Zar ne bi bilo dobro da u poruci bude i koji je taster?

Šta do sada treba uočiti – dosta nekorišćenih bita u porukama koje šalje master.



2. grupa

Posle adresne poruke, master šalje

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	W/R	?	?	?	?	?	?	?

W/R = 1 upis podatka o željenoj poziciji

W/R = 0 čitanje informacije o trenutnoj poziciji

Ako je u pitanju upis podatka o željenoj poziciji

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	1	X/Y	x	x	x	x	x	x

A nakon nje dva podataka koji nose informaciju o željenoj poziciji

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	x	x	x	D9	D8	D7	D6	D5

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	x	x	x	D4	D3	D2	D1	D0



Ako je u pitanju čitanje podatka o trenutnoj poziciji

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	X/Y	x	x	x	x	x	x

A nakon nje slejv šalje dva podataka koji nose informaciju o trenutnoj poziciji

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	x	x	x	D9	D8	D7	D6	D5

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	x	x	x	D4	D3	D2	D1	D0

Zar ne bi bilo dobro da u poruci stoji i koja je pozicija X ili Y?



Nismo razrešili maskiranje. Može slično kao u prethodnom primeru.

I za jednu i drugu grupu

Standardni format već definisan – pomeren udesno

pg – format za tu grupu

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	W/R	0	pg	pg	pg	pg	x	x

Maskiranje

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	1	1	IE	x	x	x	x	x

Čitanje statusa prekida

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	IE	AI	x	x	x	x



Promena prva grupa

adresiranje displeja

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	A3	A2	A1	A0	x	x

Čitanje tastature

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	x	x	x	x	x	x

Promena druga grupa

Zadavanje pozicije

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	X/Y	x	x	x	x	x

Čitanje pozicije

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	X/Y	x	x	x	x	x



Pouzdanost prenosa

Imamo mesta za bit parnosti ili neki drugi zaštitni kod u samim porukama.

Možemo i da pređemo na 8 bitni prenos ili ...

Smeta dekodovana poruka za displej za svaku diodu. Što ne bi slali cifru

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	D3	D2	D1	D0	x	x	x	x

Uveščemo eho protokol i timeout logiku.

Može i ACK + timeout logika.



Dosta pravolinijsko razmišljanje na osnovu prethodnog primera

Sa puno neiskorišćenih bita.

Uočiti da ako master treba da nešto prikazuje na displeju on ionako mora da zna šta je na preostalim ciframa. Kada slejv menja tu cifru koju mu je master poslalo. Očigledno je da mora da se uvede neki kriterijum inače slejv može da prikazuje „svašta“ neka međustanja. Ko zna kada će stići kompletan validan podatak.

Da se mi vratimo skroz na početak, znajući šta nam sve treba i kako smo implementirali.



Počecemo sa 9-bitnim prenosom

Adresiranje

D8	D7	D6	D5	D4	D3	D2	D1	D0
M=1	G0	A3	A2	A1	A0	-	-	-

G0 – adresa grupe

A3 .. A0 – adresa unutar grupe 0 .. 9

A3 .. A0 = 1 1 1 1 – zajednička adresa – broadcast adresa
za zajedničke akcije, npr. identifikaciju prekida

Ali da ipak ovo promenimo i odmah da razrešimo W i R kao i prekide

D8	D7	D6	D5	D4	D3	D2	D1	D0
M=1	W/R	SR	IE	G0	A3	A2	A1	A0



Maskiranje prekida Moguće i celoj grupi preko broadcast adrese

D8	D7	D6	D5	D4	D3	D2	D1	D0
M=1	1	1	IE	G0	A3	A2	A1	A0

Čitanje statusa **NIJE moguće celoj grupi preko broadcast adrese**

D8	D7	D6	D5	D4	D3	D2	D1	D0
M=1	0	1	x	x	x	x	x	x

Poruka od slejva – crveno što da ne

D8	D7	D6	D5	D4	D3	D2	D1	D0
M=0	0	1	IE	AI	x	x	x	x

Standardno adresiranje

D8	D7	D6	D5	D4	D3	D2	D1	D0
M=1	W/R	0	0	G0	A3	A2	A1	A0

Uočite da smo ovo crveno mogli da iskoristimo i za još dodatne informacije



1. grupa

Znači već znamo da li je upis u displej ili čitanje tastature

Ako je upis u displej master šalje 5 poruka po formatu

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	Ci3	Ci2	Ci1	Ci0	Cj3	Cj2	Cj1	Cj0

Za svih 10 cifara a onda i 6 poruku

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	SC7	SC6	SC5	SC4	SC3	SC2	SC1	SC0

Gde je SC7..SC0 sum check po modulu 256 svih 5 prethodnih poruka.

Dodatna zaštita.

Slejev proverava, ponovo pravi sum check i ako je isti tek onda prikazuje podatak na displeju.



1. grupa

Znači već znamo da li je upis u displej ili čitanje tastature

Ako je čitanje tastature slejv šalje 2 poruke po formatu

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	Ti5	Ti4	Ti3	Ti2	Ti1	Ti0

Za oba eventualno pritisnuta tasetra – sve nule nije pritisnut a zatim 3. poruku

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	SC7	SC6	SC5	SC4	SC3	SC2	SC1	SC0

Gde je SC7..SC0 sum check po modulu 256 dve prethode poruke.

Dodatna zaštita.

Master proverava, ponovo pravi sum check i ako je isti tek uzima u obzir informaciju.



2. grupa

Bez obzira da li je čitanje ili upis pozicije obe pozicije X i Y se šalju u istom ciklusu.

3 poruke po formatu

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	X9	X8	X7	X6	X5	X4	X3

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	X2	X1	X0	Y9	Y8	Y7	Y6

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	Y5	Y4	Y3	Y2	Y1	Y0

A onda i 4. poruku

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	SC7	SC6	SC5	SC4	SC3	SC2	SC1	SC0

Gde je SC7..SC0 sum check po modulu 256 dve prethode poruke.

Dodatna zaštita.



Pouzdanost prenosa

Sum check – dosta standardan način za zaštitu prenosa.

Ne mora eho protokol.

Mora timeout logika i na jednoj i na drugoj strani.

Timeout logiku može da aktivira i slej kada zna da mu stiže paket poruka.
Zna u kojem intervalu treba da stignu sve.

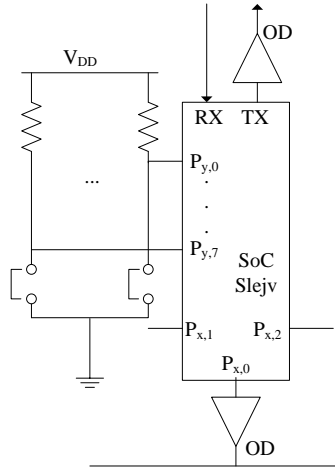


Deo jednog roka – tri grupe – 10 po grupi itd...

U trećoj grupi se nalaze uređaji u kojima se kontroliše stanje 8 digitalnih izlaza i 8 digitalnih ulaza koji se nalaze na nepoznatom naponskom potencijalu. Digitalni izlazi treba da kontrolišu digitalni signal koji predstavlja uključenje i isključenje trofaznog naizmeničnog napona 400V na uređajima sa maksimalnom strujom 100A. Ako se desi bilo koja promena digitalnih ulaza radna mesta obaveštavaju računar preko zajedničke linije za prekid INT3 (prioritet prekida određen je ulančavanjem). Zbog potreba sistema potrebno je obezbediti da računar u incidentnim situacijama može istovremeno da sve digitalne izlaze iz ove grupe postavi na pojedinačno, po svakom izlazu, softverski ili hardverski programabilnu vrednost. Računar bira da li će pojedinu grupu izlaza, u okviru istog, radnog mesta postaviti na softversku ili hardversku vrednost.

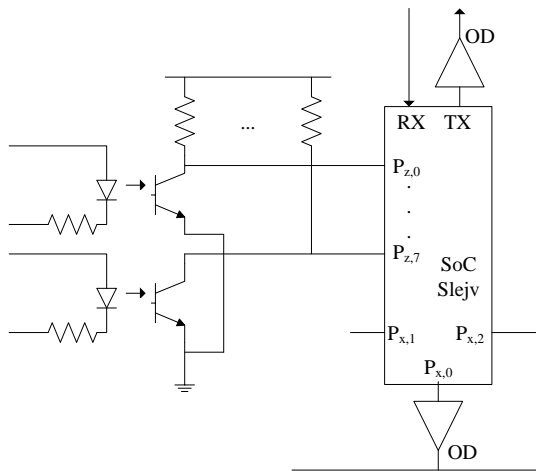


Hardverski programabilna vrednost



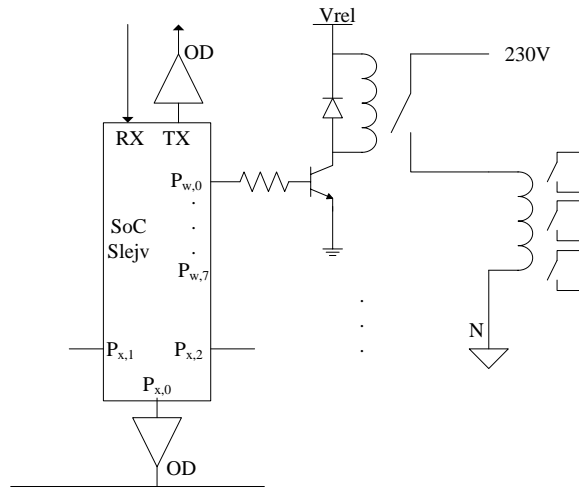
51

Galvanski razdvojeni ulazi



52

Galvanski razdvojeni izlazi
trofazni 400V, 100A



Počecemo sa 9-bitnim prenosom

Adresiranje

D8	D7	D6	D5	D4	D3	D2	D1	D0
M=1	G1	G0	A3	A2	A1	A0	-	-

G1,G0 – adresa grupe

A3 .. A0 – adresa unutar grupe 0 .. 9

A3 .. A0 = 1 1 1 1 – zajednička adresa – broadcast adresa
za zajedničke akcije, npr. identifikaciju prekida

Treća grupa ima još broadcast adresa

A3 .. A0 = 1 0 1 1 – postavi sve izlaze na hardverski programabilnu vrednost

A3 .. A0 = 1 1 0 1 – postavi sve izlaze na softverski programabilnu vrednost



A u lokalnom adresiranju
Posle adresne poruke, master šalje

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	W/R	?	?	?	?	?	?	?

W/R = 1 upis podatka na izlaze

W/R = 0 čitanje podataka o ulazima

Ako je u pitanju upis podatka

D8	D7	D6	D5	D4	D3	D2	D1	D0
0	1	A3	A2	A1	A0	x	x	x

A3 = 0 izlazi

A3 = 1 softverski programirana vrednost izlaza

Pa ide podatak itd.

Znači i ove lokalne adrese ako imamo viška možemo iskoristiti.

