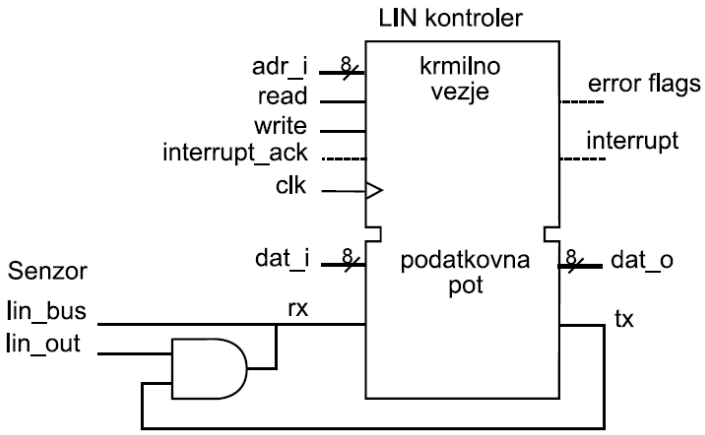
Kontrolna enota za branje senzorja na vodilu LIN

Naloga

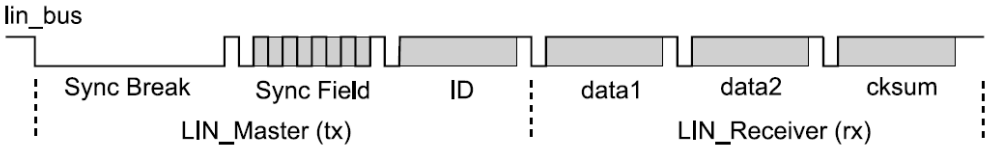
Naredi nadrejeno kontrolno enoto za branje senzorja na vodilu LIN. Krmilno vezje LIN kontrolerja naj bo prirejeno za priključitev na mikrokrmilnik PicoBlaze.



LIN kontroler ima naslovni vhod in 8 bitni podatkovni vhod in izhod. Mikrokrmilnik vidi kontroler kot množico registrov, ki jih vpisuje ali bere iz določenega naslova. Vpis ali branje določata signala read oz. write.

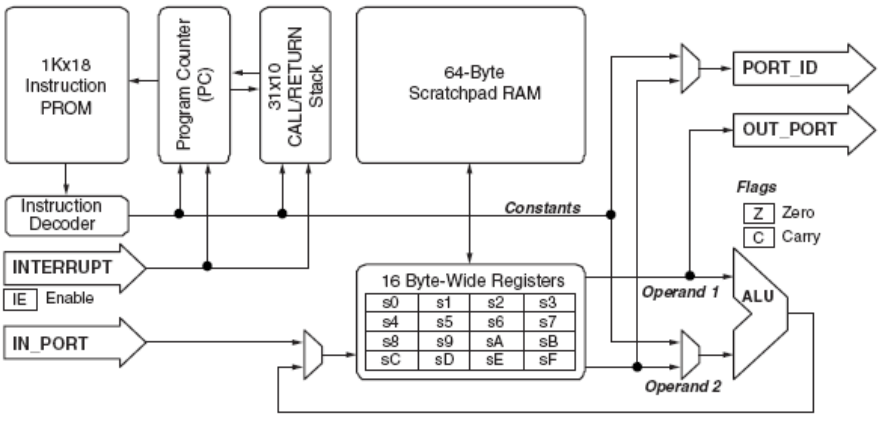
Senzor bo narejen kar znotraj prog-ramirljivega vezja, zato je priključek na vodilo LIN narejen kar z navadno IN operacijo, namesto s povezano IN logiko.

Kontrolna enota naj bo sestavljena iz enote, ki izvaja nadrejeno opravilo (LIN_Master) in podrejene sprejemne enote (LIN_Receiver), ker bomo iz senzorja le sprejemali podatke. Protokol: enota LIN_Master odda sinhronizacijo in ID, senzor nato pošlje dva podatka (data1 in data2) ter kontrolno vsoto, ki ju sprejme enota LIN_Receiver.



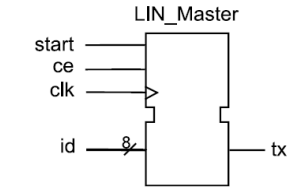
S kontrolno enoto upravlja mikrokrmilnik, ki periodično sproža oddajanje z vpisom ID na naslov 0x01, ob branju iz tega naslova pa naj kontrolna enota sporoča status (ali je podatek že sprejet). Podatek preberemo iz naslovov 0x02 in 0x03. Kot dodatno funkcionalnost lahko naredite prekinitveni signal, ki se sproži ob sprejemu podatka in zastavice za javljanje napak.

PORT_ID (adr_i)	OUT_PORT (dat_i)	IN_PORT (dat_o)
0x01	ID	status
0x02		data1
0x03		data2

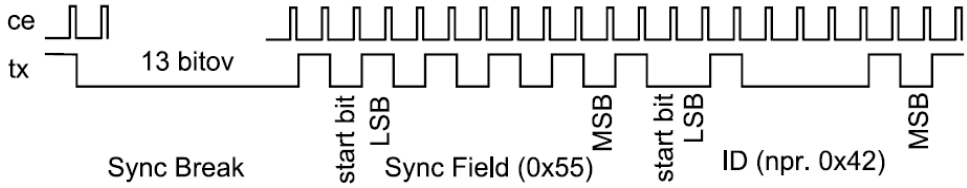


Mikrokrmilnik PicoBlaze

1. del: enota LIN_Master



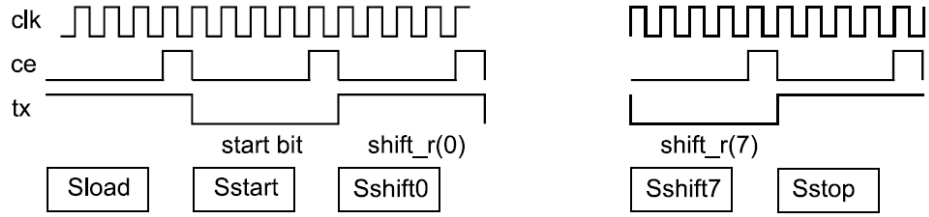
Vezje LIN_Master izvaja nadrejeno opravilo po protokolu LIN, to je oddajni proces. Vezje naj ob aktiviranju signala start odda: 13 bitov logične '0' (Sync Break), 1 stop bit (logična '1'), sinhonizacijsko kombinacijo (start bit + 0x55 + stop bit) in ID (npr. start bit + 0x42 + stop bit).



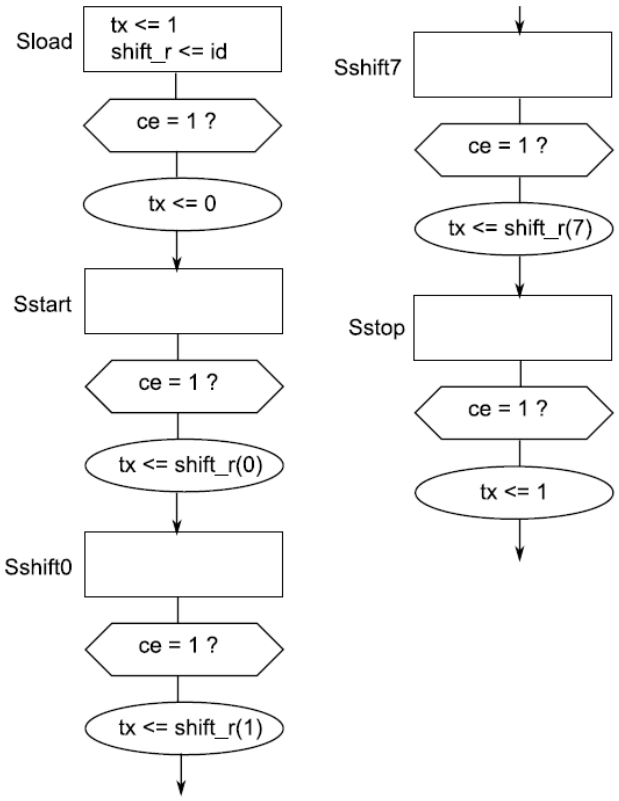
Frekvenca sistemske ure je običajno precej višja od bitne frekvence LIN vodila, zato imamo v vezju signal ce (clock enable), ki označuje bitno periodo.

1.a) oddajnik paketov po protokolu 8N1

Začnimo z vezjem, ki odda paket podatkov v obliki 8N1: en startni bit, 8 podatkovnih bitov in stop bit. Vezje naj ob aktiviranju signala start izvede algoritem prenosa podatka, ki je na vhodu id.



Prenos podatkov lahko obravnavamo kot zaporedje stanj, ki se spreminjajo ob aktivem signalu ce. V stanju Sload naložimo podatek v pomikalni register (shift_r), v stanju Sstart pošljemo startni bit (logično '0'), nato pa v stanjih Sshift0 do Sshift7 zaporedne bite iz registra. Delovanje ponazorimo z algoritmičnim diagramom prehajanja stanj (Algorithmic State Chart):




```
if rising_edge(clk) then
  case state is
    when Sload =>
      tx <= '1';
      shift_r <= id;
      if ce='1' then
        tx <= '0';
        state <= Sstart;
      end if;

    when Sstart => ...
```

Algoritmični diagram je sestavljen iz pravokotnikov, ki predstavljajo stanja. V njih so napisane prireditve, se izvršijo v določenem stanju. Prehod v naslednje stanje določajo pogoji v odločitvenih blokih (šestkotniki). Elipse predstavljajo pogojne izhodne bloke – stavke, ki se izvršijo ob prehodu v naslednje stanje. Pogojni izhodni bloki so vedno narisani za odločitvenimi bloki, na katere se nanašajo.