

Seminarska naloga pri predmetu ONT: Vmesnik za standard RS-232

Mattia Petroni, Blaž Parkelj, Martin Kranjc, Simon Rangus

University of Ljubljana, Faculty of Computer and Information Science,
Ljubljana, Slovenia

Abstract. Seminarska naloga obravnava implementacijo vmesnika za standard RS-232 s kvantnim celičnim avtomatom. S pomočjo programa QCADesigner smo implementirali oddajnik in sprejemnik. Nato smo oba modela uporabili za izdelavo novega modela, v katerem smo izhod oddajnika pripeljali na vhod sprejemnika. Na koncu smo za te tri modele pognali simulacijo z vhodnimi vektorji, ki smo jih določili v skladu s standardom RS-232.

Keywords. kvantni celični avtomat, standard RS-232, RS-232 sprejemnik, RS-232 oddajnik

Contents.

1 Uvod	2
2 Metode	2
2.1 Standard RS 232	2
2.2 Start-Stop način prenosa	3
2.3 UART vmesnik	4
3 Rezultati	5
3.1 Oddajno vezje	5
3.2 QCA RS-232 oddajnik	6
3.3 Sprejemno vezje	7
3.4 QCA RS-232 sprejemnik	8
3.5 QCA model RS-232 standarda	9
4 Zaključek	10
References	11

1. Uvod

RS-232 standard je v rabi v računalniškem svetu že 40 let. Njegova preprostost izvedbe, prenosljivost in adaptacija na fizičnem nivoju na vse danes uporabljene napajalne linije je skrivnost njegovega uspeha v tem dolgem obdobju. Danes je zaradi velike spremembe v hitrosti diskov, glavnih pomnilnikov in predvsem mikroprocesorjev postal prenos podatkov najbolj pereč problem v računalniških sistemih. Ozko grlo predstavlja prav prenos podatkov po linijah, po katerih zaradi zakasnitev in drugih elektrotehničnih problemov žal ne moremo prenašati ogromnih količin podatkov s hitrostjo mikroprocesorjev. Novejša optična tehnologija pa je že sposobna delovati na takih hitrostih, toda njegova uporaba znotraj računalniških komponent je še v fazi raziskovanja in se le ta uporablja za druge namene, npr. telefonija, IP televizija, itd, predvsem pa za prenos podatkov na velikih razdaljah. Standardi z višjo hitrostjo prenosa, kot so USB 2.0 in USB 3.0, so danes v veliki meri zamenjali standard RS-232. Megabitne hitrosti, ki jih danes dosežejo USB vmesniki, so nedosegljive za RS-232 vmesnike. Iz tega lahko sklepamo, da bo ta standard počasi "šel v pokoj".

S QCA tehnologijo in s sledečo seminarsko nalogo želimo predstaviti, bolj kot zgolj izboljšavo, posebno implementacijo vmesnika tega standarda, ki bi zaradi njenih poenostavitev in preprostosti teoretično lahko še konkuriral. V nadaljevanju bomo pojasnili te značilnosti ter predstavili logično izvedbo in njeno implementacijo s QCA celicami.

2. Metode

2.1. Standard RS 232

Povezovalni standard RS-232, imenovan tudi V.24, je bil vpeljan leta 1969 pri organizaciji za telekomunikacijske standarde ITU-T (oziroma nekdanji CCITT).

Standard opisuje digitalno serijsko asimetrično komunikacijo med oddajnikom in sprejemnikom. Kot že beseda "digitalen" pove, imamo dva podatkovna nivoja: '0', ki je določena z napetostjo med $-5V$ in $-15V$ in '1', z napetostjo med $5V$ in $15V$. To so napetosti oddajnika, medtem ko so na sprejemni strani detektirane kot '0' vse vrednosti, ki so manjše od $-3V$, in '1' vse vrednosti, ki so večje od $3V$. Iz teh podatkov lahko razberemo, da je odpornost proti šumu vsaj $2V$.

Glede na to, da standard nima nobenega zagotovila glede odbojev in presluhov, mora biti hitrost spreminjanja signala (ob preklopih) omejena z $30V/\mu s$ (to je v bistvu tudi edini ukrep proti odbojem in presluhom). To je zelo pomemben podatek, ki nam pove, da mora biti čas vzpona t_r (raise-time) ali padca t_f (fall-time) omejen: $333ns < t_r < 1000ns$. Med delovanjem čas vzpona signalov doseže vrednosti, ki so prav v tem rangu. Ta čas nas pripelje do maksimalne dolžine linije, ki znaša $15m$.

$$\frac{t_r}{T} \geq 5, \frac{t_r}{l * \delta} \geq 5 \Rightarrow l_{max} = \frac{t_{rmin}}{5 * \delta} = 15m$$

kjer je T čas potovanja signala, l dolžina linije, δ zakasnitev na enoto dolžine, ki tipično znaša $\delta = 4.5ns/m$, in t_{rmin} znaša $333ns$. Torej do 15 metrov omenjena hitrost preklopa signalov zagotavlja pravilen prenos.

Po določilih standarda je lahko signal med nizkim in visokim stanjem (to je med $-5V$ in $5V$) največ 4% urine periode. Ta podatek nas pripelje do maksimalne hitrosti,

s katero bi lahko pošiljali podatke v idealnem primeru, ki znaša $40kHz$, vendar standard zaradi disperzije in slabljenja le to omejuje na $20kHz$. Hitrost $40kHz$ bi v principu lahko tudi dosegli, če bi uporabili krajšo linijo, kar nam skrajša tudi čas vzpona oziroma padca. Glede na to, da so hitrosti preklopa signalov omejene, je povsem logično sklepati, da je za to potrebno v vmesnikih RS-232 uporabljati ustrezen strojno opremo, ki zna hitre mikroprocesorske signale ustrezno zakasnit v času preklopa iz enega logičnega stanja v drugo. V nadaljevanju bomo videli, da je to v primeru QCA tehnologije povsem odveč in da lahko deluje prenos podatkov s frekvenco ure QCA celic.

2.2. Start-Stop način prenosa

Start-Stop je asinhronski način prenosa podatkov, ki se je prvič pojavil leta 1914 z napravami, ki jim pravimo teleprinterji. Pri prenosu pošiljamo znake, pri čemer je vsak znak sestavljen iz treh delov: start in stop bita ter podatkovnih bitov (vključno s paritetnim bitom). Iz te strukture pošiljanja izhaja tudi ime Start-Stop.

Mirovno stanje linije je običajno označeno z vrednostjo '1'. Start bit, ki sprejemniku sporoči začetek znaka, je vedno različen od tega stanja. Start bitu sledijo podatkovni biti, ki jim lahko sledi (ni pa obvezno) paritetni bit. Ti biti določajo znak, ki se pošilja, dolžina tega pa je lahko različna, odvisno od abecede, ki jo uporabimo. Po CCITT predpisih se na linijo najprej pošlje bit z najmanjšo težo, zadnji pa bit z največjo težo. Ta predpis upoštevajo vsi mikroprocesorski komunikacijski adapterji.

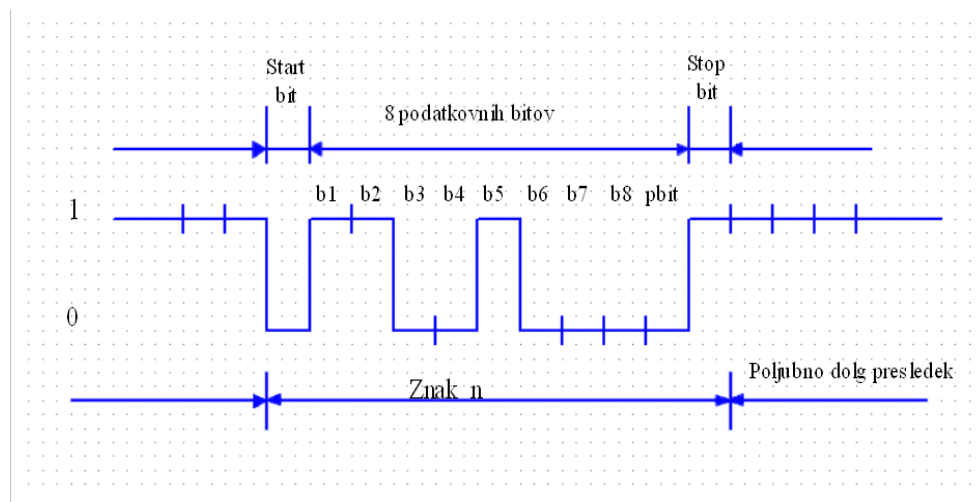


Figure 1. Asinhroni Start-Stop način prenosa (povzeto po [3]).

Stop bit, ki sledi znakovnim bitom, je vedno enak mirovnemu stanju linije. Včasih imamo tudi dva stop bita. Stop bit zagotavlja, da bo pred naslednjim znakom linija v mirovnem stanju najmanj en bitni interval. Lahko mu sledi naslednji znak takoj, lahko pa je vmes poljubno dolg presledek, v katerem je linija v mirovnem stanju.

2.3. UART vmesnik

UART (Universal Asynchronous Receiver Transmitter) je skupno ime za elemente, ki jih danes ponuja večina proizvajalcev mikroprocesorjev za asinhronske komunikacije. Vsak UART vsebuje oddajnik in sprejemnik, ki sta med seboj neodvisna in lahko delujeta paralelno.

Hitrost oddaje in sprejema podatkov je določena z uro. Danes se predvsem uporablja Baud-*n*e generatorje ure. Standardne hitrosti prenosa so od 50 in 75 pa vse do 9600 in 19200 bitov na sekundo. Te hitrosti so pomembne pri komunikacijah in pri povezovanju vhodno-izhodnih naprav, ker te uporabljajo samo standardne hitrosti. Znotraj svojega sistema pa lahko uporabimo katerokoli hitrost. Vsak UART čip je mogoče sprogramirati. S programom določimo število podatkovnih bitov, paritetni bit, število stop bitov, prekinitve ob oddaji in sprejemu, itd. UART pri oddaji vsak znak pridobljen od procesorja pretvori v prenosni format, za katerega je sprogramiran in obratno pri sprejemu, hkrati pa preveri tudi pariteto in zgradbo zaporedja bitov (start, število podatkovnih bitov, stop).

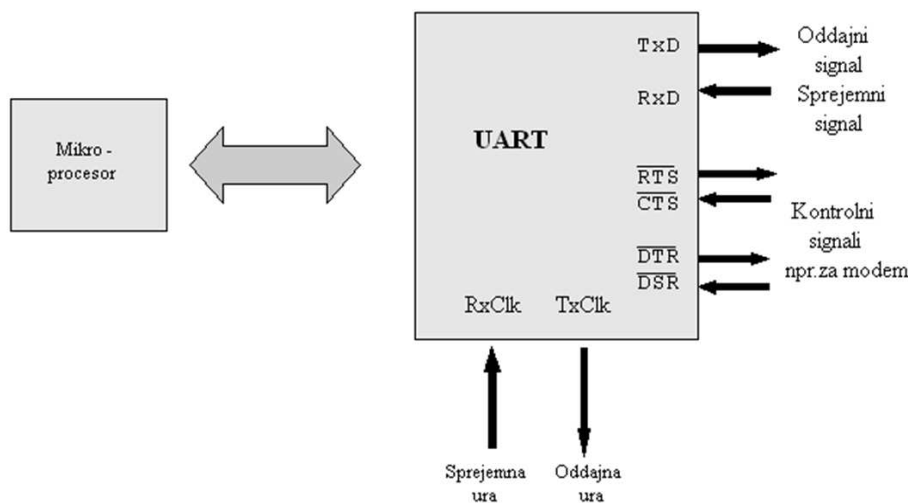


Figure 2. Tipičen UART in njegovi signali [3].

Kratek opis kontrolnih signalov iz slike Fig. 2

- RTS : request to send (signal, ki pove sprejemniku, da je oddajnik pripravljen na prenos)
- CTS : clear to send (naprava je uspešno prebrala podatke)
- DTR : data terminal ready (postavi se v nizko stanje ob preklopu naprave)
- DSR : data set ready (podobno kot DTR – to sta znaka, da napravi delujeta)

Pri oddaji in sprejemu se uporablja t.i. dvojno izravnavanje. Sprejemnik ima dva registra: sprejemni pomikalni in sprejemni podatkovni. Ko je celoten znak sprejet v pomikalni register, se podatkovni biti prenesejo v sprejemni podatkovni register. Od tod je dostopen procesorju. Čas, v katerem mora procesor prebrati znak, je enak trajanju dolžini znaka v bitnih intervalih. Če v tem času podatka ne prebere in se

medtem sprejme nov znak, je prejšnji podatek izgubljen. Temu pravimo “overrun”. Podobno je pri pošiljanju. Ko procesor vpiše znak v oddajni podatkovni register, se ta premakne v oddajni pomikalni pod pogojem, da je prost, in pošiljanje se začne bit za bitom. Medtem ko se biti oddajajo, lahko procesor vpiše v podatkovni register nov znak, ki čaka, da se pomikalni register izprazni.

3. Rezultati

Pri gradnji logike vezja RS-232 za oddajnik in sprejemnik start-stop protokola smo se odločili za izdelavo dveh različnih qca datotek. V eni je implementiran samo oddajnik, v drugi pa samo sprejemnik. Datoteki smo na koncu uvažali kot bloke in sestavili skupno testno vezje za simulacijo prenosa podatkov skozi oddajnik in sprejemnik.

Pri izvedbi s kvantnimi celičnimi avtomati smo poenostavili logično strukturo vmesnika na samo 2 podatkovna bita brez paritete. Taka izbira povzroči, da vezje ne postane preveliko in posledično, da simulacija ne traja predolgo. Pri sprejemniku je pri povečanju števila podatkovnih bitov dovolj, da sekvenčno strukturo multiplekserja in relativne D pomnilne celice ponavimo tolikokrat, kolikor je podatkovnih bitov. Pri sprejemniku je stvar podobna. Potrebno je dodati toliko izhodov, kolikor je podatkovnih bitov.

3.1. Oddajno vezje

Slika 2.1.1 ponazarja logično vezje oddajniškega čipa. Pri tem želimo pojasniti, da predstavlja le bločno shemo vezja. Razlike v tehnologiji CMOS oz. QCA so velike. V to shemo namenoma nismo narisali urin signal, kajti za primer QCA tehnologije je le ta nepotreben.

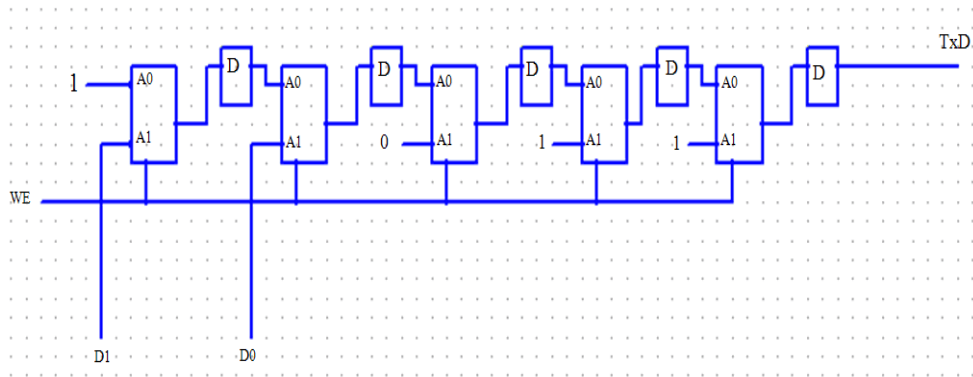


Figure 3. Logično vezje oddajnika.

Ko je signal WE v visokem stanju (impulz), vsi multiplekserji dajo na izhod (in torej na vhod v pomnilno celico) vrednost A1. V nasprotnem primeru pa multiplekserji dajo na izhod vrednosti A0. To so vrednosti, ki se hranijo v D-pomnilnih celicah. Delovanje vezja je pri impulzu WE torej sekvenčno. V 6 urinih periodah se torej prenesejo vsi obvezni kontrolni in podatkovni biti skozi linijo na izhod TxD:

- dve obvezni enici, ki tvorita IDLE znak (obvezen čas med dvema prenosoma = $204\mu s$ [1])

- ničlo, ki predstavlja start bit (določa začetek podatkovnega niza)
- v našem primeru 2 podatkovna bita D0 in D1
- končno enico, ki predstavlja stop bit (določa konec podatkovnega niza)

Vidimo, da vezje postavlja na linijo TxD same enice, ko je WE v nizkem stanju in je prenos končan. Te enice zagotavlja prvi multiplekser na vhodu A0. S tem je zagotovljen princip, po katerem protokol start-stop lahko enolično določa začetek in konec prenosa. Poleg tega pa je zagotovljen standard RS-232, ki pravi, da sta liniji TxD in RxD (ti pa predstavljata isto linijo) v času mirovanja vedno v visokem stanju. Oddajniški čip je torej sestavljen iz petih preprostih 2/1 MUX in petih D-pomnilnih celicah. Nekatere multipleksorje lahko v primeru konstantnih vhodov poenostavimo. To je primer tretjega multipleksorja. MUX 2/1 s konstantnim vhodom 0 lahko reduciramo na enostavnejša AND vrata:

$$x = A_1 \wedge WE \vee A_0 \wedge \overline{WE} = A_1 \wedge WE \vee 0 \wedge \overline{WE} = A_1 \wedge WE$$

Tako poenostavitev smo uporabili v qca vezju. Potrebno je poudariti, da ta rešitev (za oddajnik) s kaskadno vezavo multipleksorjev ni najbolj primerna za realizacijo v tehnologiji CMOS. V QCA tehnologiji pa takšna vezava multipleksorjev predstavlja dobro rešitev.

3.2. QCA RS-232 oddajnik

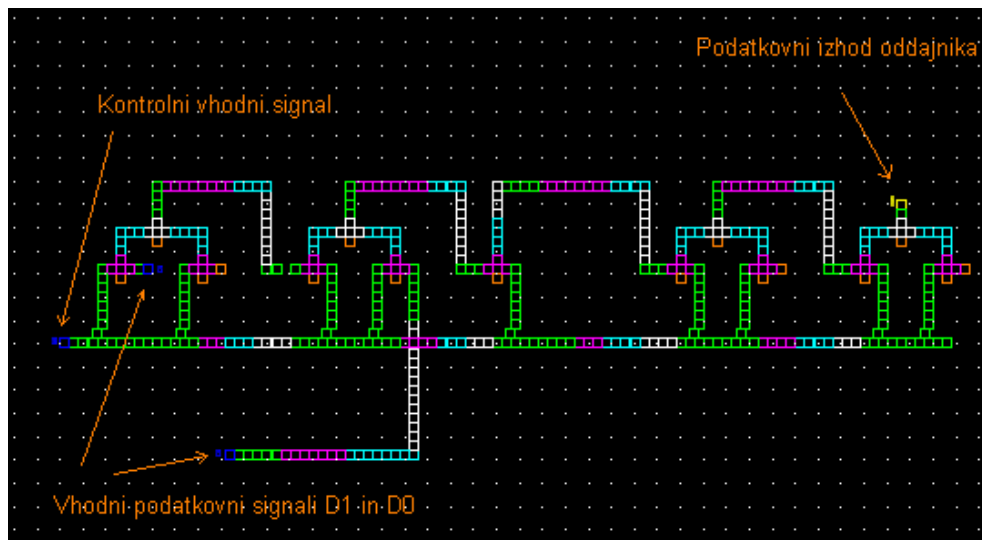


Figure 4. QCA model za oddajnik.

Pri implementaciji QCA modela oddajnika smo izhajali iz logične sheme iz slike Fig. 3. Tako smo dobili QCA strukturo, ki je prikazana na sliki Fig. 4. Multipleksorji in D pomnilne celice iz logične sheme so realizirane z običajnimi kvantnimi celicami. Kontrolni signal WE je pa realiziran s kvantnimi celicami z notranjo rotacijo za 45° stopinj. Ker pa nam 45 stopinjska linija omogoča križanje linij, smo lahko vezje realizirali na eni plasti.

kjer sta xx podatkovna bita D0 in D1. Takrat sta podatka veljavna in lahko ju posredujemo naprej. Pri logični shemi sprejemnika velja opozoriti na malenkost, da je izhod iz D4, ki gre v AND vrata, negiran.

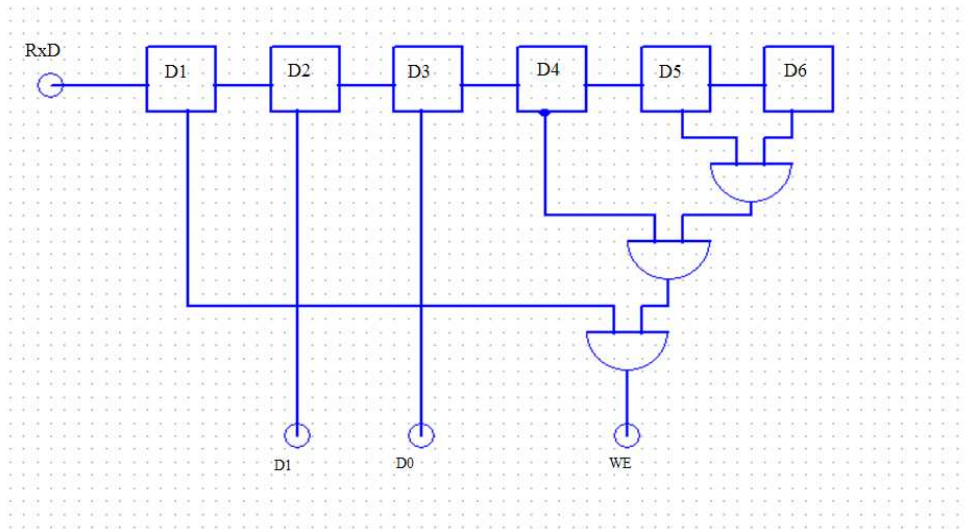


Figure 7. Logično vezje sprejemnika.

3.4. QCA RS-232 sprejemnik

Sprejemniški čip realiziran s QCA celicami je prikazan na sliki Fig. 8. Vidimo, da je ta struktura zelo enostavna. V celem modelu imamo tri AND vrata in en negator. Navpični liniji, na katerih dobimo podatkovna bita, sta 45 stopinjski liniji. Tako smo lahko naredili križanje linij v eni sami plasti. Vse ostale celice nimajo 45 stopinjske rotacije.

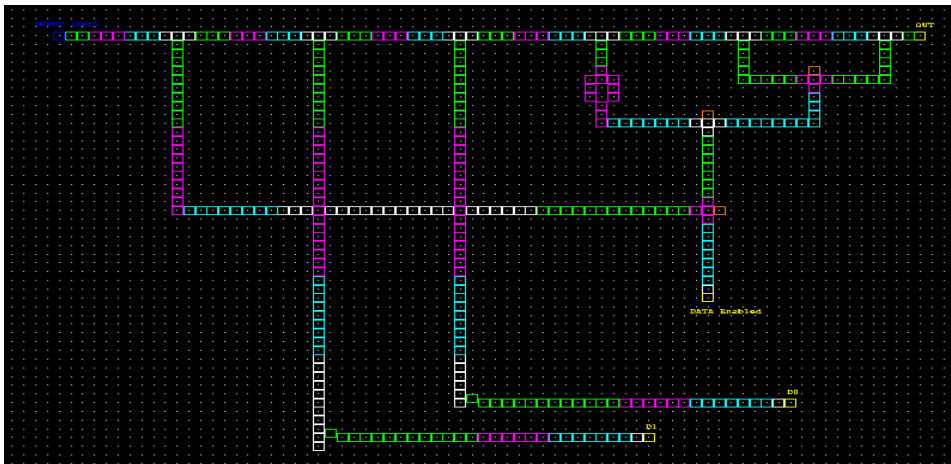


Figure 8. QCA model sprejemnika.

Za simulacijo smo vzeli vhodne vektorje iz slike Fig. 9 in rezultat teh vhodov je prikazan na sliki Fig. 10. Izhodi so veljavni samo takrat, ko je DE v visokem stanju. DE signal se postavi po 6ih urinih periodah od prejetega start bita (pri pogoju, da detektiramo tudi stop bit), oziroma 3 urine periode od prejetega stop bita. Signal OUT je samo pomožni izhod, ki nam je služil samo za to, da smo videli, če se je vhod pravilno prenesel skozi linijo qca celic.

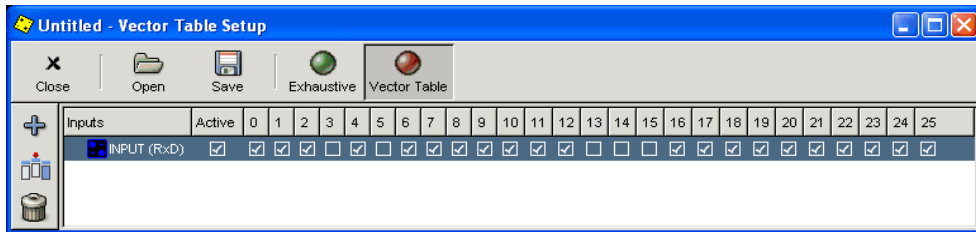


Figure 9. Vhodni vektor za sprejemnik.

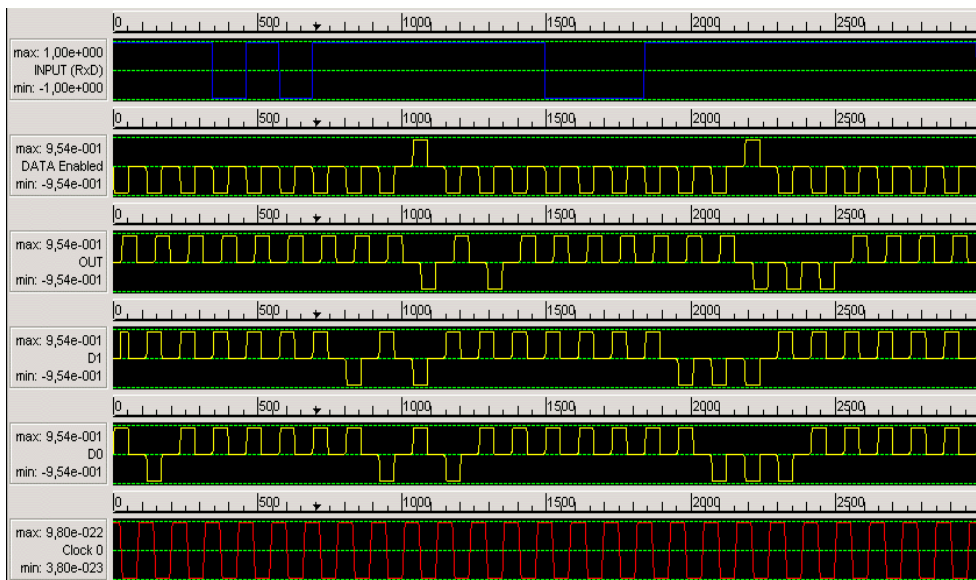


Figure 10. Rezultat simulacije za vhodne vektorje iz slike Fig. 9.

3.5. QCA model RS-232 standarda

Oddajnik in sprejemnik smo najprej izvozili kot bloka. Nato smo naredili novo vezje, v katerega smo oba bloka uvozili. Nato smo izhod oddajnika povezali na vhod sprejemnika. Rezultat je bil (po vztrajnem delu s QCADesigner-jem) delujoč sistem za prenos podatkov skozi serijski vmesnik RS-232. Sistem vsebuje več kot 500 celic. Pri tem modelu simulacija tipa bistabilne aproksimacije ne deluje pravilno, čeprav vsak od obeh blokov deluje pravilno. Zato smo simulacijo pognali s koherentnim računanjem, ki pa je trajala v našem primeru več kot 15 minut (na procesorju Intel Quad Core 2.7Ghz). Sistem ima:

- 2 podatkovna vhoda
- WE (write-enable) vhodni kontrolni signal za multipleksorje v oddajnem čipu
- 2 podatkovna izhoda
- izhod sprejemnega pomikalnega registra (za kontrolo)
- DE (data-enable) signal v sprejemniškem čipu, ki pove, kdaj so podatki na izhodu veljavni

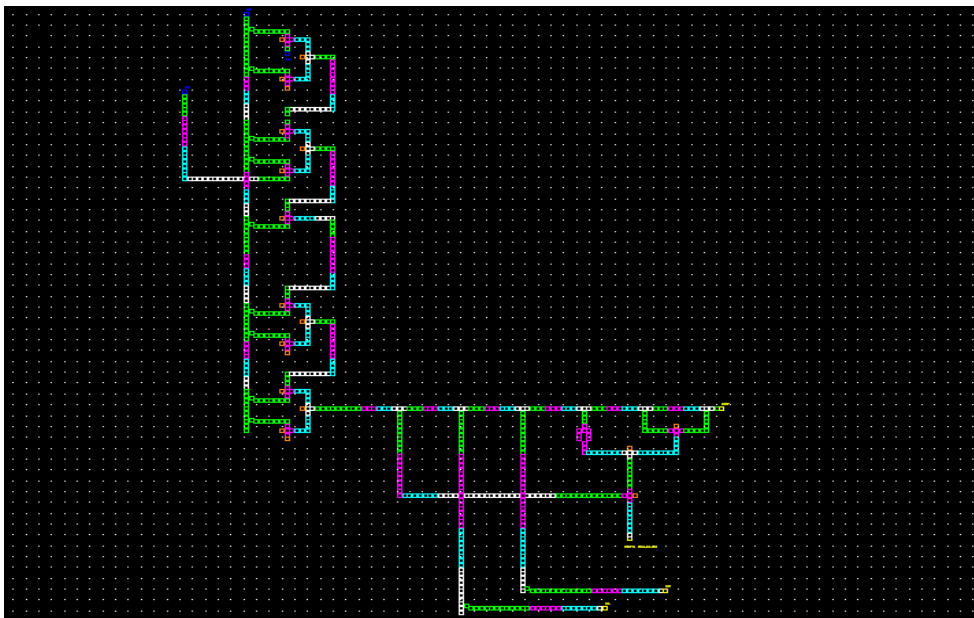


Figure 11. QCA model start-stop protokola v vmesniku RS-232.

Iz logičnega vidika, je ta sistem skladen s povezavo logičnih shem iz slik Fig. 7 in Fig. 3. Se pravi iz kvantitativnega vidika, je sistem podan z 12 AND vrat, 4 OR vrat in samo treh linijskih križanj. Primer simulacije je prikazan na sliki Fig. 12.

4. Zaključek

Iz logičnih realizaciji oddajnika in sprejemnika v QCADesignerju je razvidno, da so postali kontrolni signali UART čipa RTS in CTS nepotrebni. Toda paziti moramo na to, da realiziran oddajniški vmesnik v QCA zahteva, da se prenosi izvedejo postopoma z minimalno zakasnitvijo 6 urinih period. Ta čas je potreben zato, da se zadnji podatkovni bit lahko pravilno prenese skozi TxD. Takoj za tem pa impulz kontrolnega signala multiplekserjev (WE) pride do zadnjega MUX-a in ta pa prepusti skozi prvi bit idle znaka. Pojasniti pa moramo še, da sistem deluje le v primeru, če je WE signal enotin impulz in da so posamezni znaki ustrezno zakasnjeni. V naših primerih smo ti dve zahtevi lahko enostavno dopolnili v času simulacije z nastavitvijo pravih vhodnih vektorjev, ki so med seboj oddaljeni vsaj 6 urinih period. Če ti dve zahtevi ne bi dopolnili v simulaciji, potem bi morali realizirati še signala RTS in CTS. Ker bi tako vezje postalo zelo zapleteno za realizacijo v eni sami plasti pa tudi zelo malo berljivo in neurejeno, smo se enotno odločili za rešitev preko simulacije.

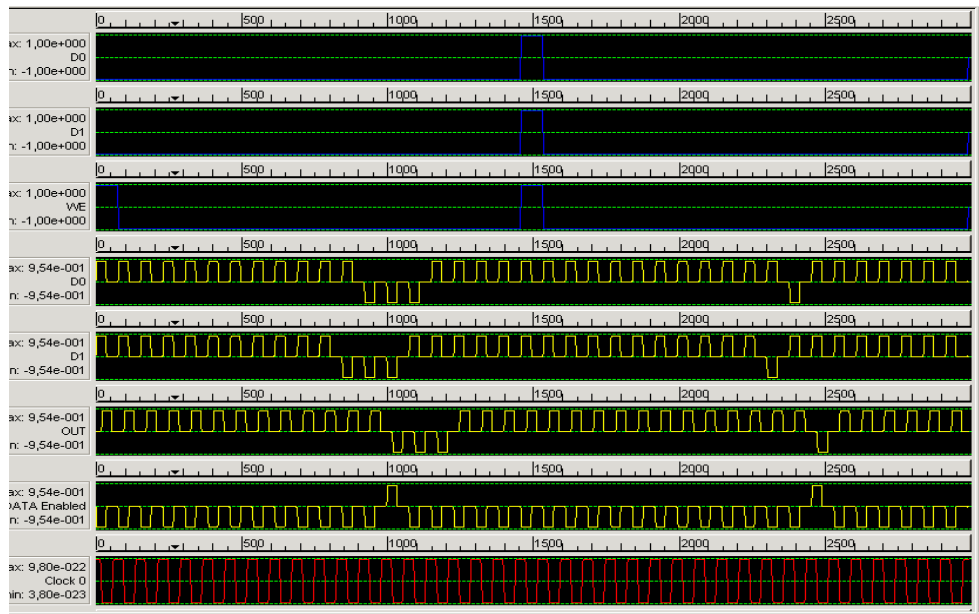


Figure 12. Rezultati simulacije za QCA model standarda RS-232.

References

- [1] http://en.wikipedia.org/wiki/RS_232
- [2] Zapiski iz predavanj prof. Dušana Kodeka predmeta Vhodno-Izhodni sistemi in naprave, 2009, Mattia Petroni
- [3] D. Kodek. Mikroprocesorski sistemi. (c) DiTim d.o.o., Ljubljana, 1993