2
Digitalni signali

2.1 Binarni signali

Digitalni signali so signali, ki lahko zavzamejo le končno število različnih stanj. Imenujemo jih tudi *diskretni signali*. Električni signali, ki predstavljajo fizikalne količine, so večinoma analogni in lahko zavzamejo poljubno vrednost znotraj nekega območja. Primer takšnega signala je napetost, ki jo dobimo iz temperaturnega senzorja. Vrednost temperature je realno število z neomejenim številom decimalnih mest (npr. $20.213657...°C$). Preprost primer digitalnega signala je število dvignjenih prstov na roki - število je lahko le ena izmed vrednosti iz diskretnega območja med 0 in 10. Beseda *digitalni* prihaja iz latinskega izraza digitus, ki pomeni prst. V digitalnih elektronskih vezjih se največkrat uporabljajo *binarni* signali, ki lahko zavzemajo le dve možni stanji označeni kot:

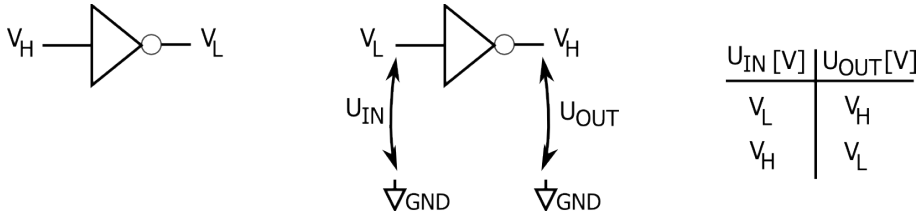
- napačno (false) ali pravilno (true),
- nizko (potencial V_L) ali visoko (V_H),
- številka 0 ali 1.

Dve stanji opisujeta preproste pojave, kot so prižgana oziroma ugasnjena žarnica ali stikalo. Predstavljata lahko logično trditev, ki je pravilna ali napačna. S količinami v digitalni obliki izvajamo osnovne logične operacije. Primer preproste logične operacije je negacija: napačno stanje spremenimo v pravilno in obratno. Element, ki izvaja logično negacijo imenujemo negator ali inverter.

Nizko in visoko stanje sta v vezju predstavljena z nizkim (V_L) in visokim (V_H) potencialom signala oziroma nizko in visoko napetostjo proti masi. Slika 2.2 predstavlja potenciale na inverterju pri visokem in pri nizkem stanju na vhodu. Delovanje preprostih logičnih elementov na

Slika 2.1: Logični negator obrne vrednost signala na vhodu

kratko opišemo s tabelo, kjer podamo stanja izhodov pri vseh možnih stanjih vhodov (negator ima le en vhod in torej le dve možni stanji).

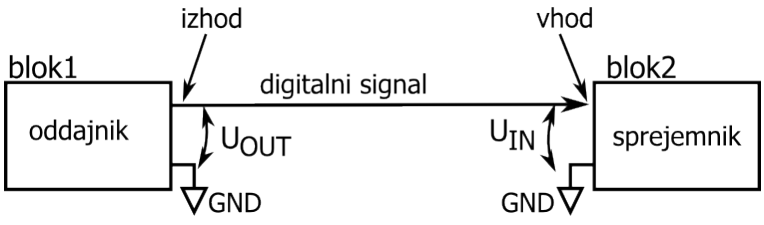


Slika 2.2: Potenciali in napetosti na vhodu in izhodu negatorja

Najbolj pogost zapis digitalnih stanj je v obliki številskih vrednosti 0 in 1. Takšen zapis ni samo najkrajši, ampak je tudi primeren za računanje, saj digitalna vezja velikokrat izvajajo računske operacije. Vrednost 0 ali 1, ki jo zavzame enostaven signal, imenujemo binarna števka ali **bit** (angl. binary digit). Vodila v digitalnih vezjih pa prenašajo večbitne vrednosti v dvojiškem zapisu.

2.2 Statični red

Osnovni gradniki digitalnih vezij se obnašajo kot preprosta elektronska stikala, ki preklapljajo med potencialom V_L in V_H. Elektronska stikala so bila včasih narejena z releji ali elektronkami, danes pa z različnimi elementi v polprevodniški tehnologiji. Z razvojem elektronike se spreminjajo tudi osnovni elementi in njihove električne lastnosti. Da bi digitalna vezja v različnih tehnologijah lahko povezali med seboj, moramo uvesti nek dogovor, ki določa potenciale za nizko in visoko stanje na vhodih in izhodih vezja. Vrednosti potencialov oz. napetosti gledamo v ustaljenem stanju, zato se dogovor imenuje statični red (angl. static discipline).



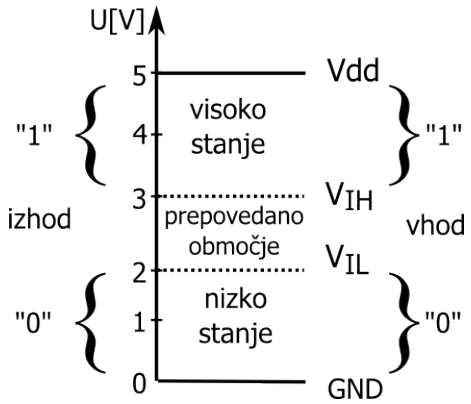
Slika 2.3: Povezava dveh digitalnih blokov

Elektronska stikala v digitalnih vezjih niso idealna, imajo neko upornost, ki povzroči da visoko stanje V_H ni enako napajalnemu potencialu Vdd in da je nizko stanje V_L nekoliko višje od

potenciala mase. Prav tako moramo upoštevati možne razlike v napajalnih napetostih digitalnih gradnikov. Statični red omogoča pravilno interpretacijo signalov, ki potujejo med dvema digitalnima blokoma. Vzemimo najbolj preprost in pogost primer digitalne povezave, ko je izhod enega bloka vezan na vhod drugega bloka, kot prikazuje slika 2.3. Enostaven dogovor bi lahko določal, da predstavljajo vse napetosti med Vdd in Vdd / 2 visoko stanje (logično "1"), napetosti med 0 in Vdd / 2 pa nizko stanje (logično "0"):

$$\begin{array}{ll} \text{logična "0":} & 0V \leq V_L \leq Vdd/2 \\ \text{logična "1":} & Vdd/2 \leq V_H \leq Vdd \end{array}$$

Pri takšnem dogovoru se pojavi težava, če dobi sprejemnik na vhod napetost Vdd / 2. Da bi lahko sprejemnik nedvoumno razločeval med logično "0" in "1" dodamo prepovedano območje potencialov na signalni liniji, kot prikazuje slika 2.4. Vpeljali smo dva nova potenciala: V_{IH} je minimalni potencial, ki se na vhodu logičnega vezja interpretira kot visoko stanje, V_{IL} pa maksimalni, ki predstavlja nizko stanje.

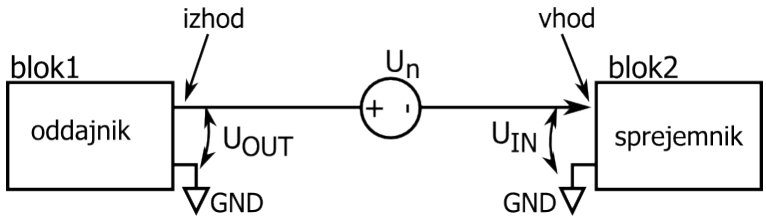


Slika 2.4: Dogovor o potencialih za nizko in visoko stanje

Konkretne vrednosti so odvisne od tehnologije in zahtev - večje prepovedano območje poveča robustnost sistema, večje območje pravilnih stanj pa združljivost z več tehnologijami. Za neko 5V CMOS tehnologijo, bi lahko uporabili vrednosti:

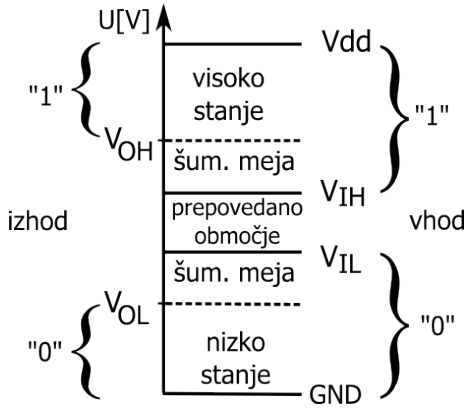
$$\begin{array}{ll} \text{logična "0":} & 0V \leq V_L \leq 2V \\ \text{logična "1":} & 3V \leq V_H \leq 5V \end{array}$$

Na signalni povezavi se lahko inducira šum, ki ga modeliramo kot dodatno napetost U_n, kot prikazuje slika 2.5. Šum lahko povzroči neveljavno stanje na vhodu sprejemnika, kljub temu da je na oddajni strani stanje z veljavnim potencialom. Inducirani šum je v splošnem pozitivna ali negativna napetost. Denimo, da se na povezavi inducira $U_n = 0.5V$ šumne napetosti. Če je na izhodu CMOS vezja nizko stanje z napetostjo $1V$, se bo vsota napetosti $1V + 0.5V = 1.5V$ na vhodu vezja še vedno pravilno obravnavala kot logična "0". Če pa je ne izhodu vezja še vedno veljavna vrednost $2V$, bo vsota $2V + 0.5V = 2.5V$ v prepovedanem območju.



Slika 2.5: Na povezavi dveh digitalnih blokov je šum, ki povzroči neveljavno stanje na sprejemniku

Odpornost na šum naredimo tako, da določimo ožje območje veljavnih potencialov na oddajni strani in širše na sprejemni strani, kot prikazuje slika 2.6. Razlika v širini območja se imenuje šumna meja in zagotavlja določeno odpornost na induciran šum pri komunikaciji.



Slika 2.6: Statični red z upoštevanjem šumne meje

Veljavno območje potencialov na oddajni in sprejemni strani je sedaj podano z enačbami:

$$\begin{array}{lcc} & oddajnik & sprejemnik \\ \text{logična ”0”:} & 0V \leq V_L \leq V_{OL} & 0V \leq V_L \leq V_{IL} \\ \text{logična ”1”:} & V_{OH} \leq V_H \leq Vdd & V_{IH} \leq V_H \leq Vdd \end{array}$$

Vrednosti potencialov določajo standardi. Izjava proizvajalca digitalnih integriranih vezij o skladnosti s standardom zagotavlja, da bomo brez težav povezali signale različnih vezij med seboj. Tabela 2.1 prikazuje vrednosti potencialov iz standarda JEDEC JESD8C za digitalna signale v 3.3V tehnologiji LVCMOS.

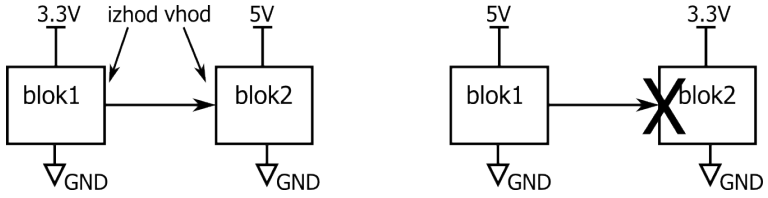
Inducirani šum lahko povzroči, da bo vrednost signala izven meja napajalnih napetosti: višja od Vdd ali nižja od GND (negativna napetost). Takšen signal se na sprejemniku sicer pravilno interpretira, lahko pa povzroči uničenje vezja, če preseže določene meje.

oznaka	pomen	min	max
Vdd	napajalna napetost	$3.0V$	$3.6V$
V_{IH}	vhodni visok nivo	$2V$	$Vdd + 0.3V$
V_{IL}	vhodni nizek nivo	$-0.3V$	$+0.8V$
V_{OH}	izhodni visok nivo	$Vdd - 0.2V$	
V_{OL}	izhodni nizek nivo		$+0.2V$

Tabela 2.1: Statični parametri 3.3V tehnologije LVCMOS pri normalnem razponu Vdd (JEDEC)

2.2.1 Povezovanje različnih standardov

Včasih potrebujemo povezavo med digitalnimi bloki, ki uporabljajo različne standarde. Poglejmo si primer povezave 5V CMOS in 3.3V LVCMOS bloka. Izhod 3.3V bloka lahko brez težav krmili vhod 5V bloka, kot prikazuje leva stran slike 2.7. Visoko stanje LVCMOS je vsaj 3.1V, kar je dovolj velik potencial, da se interpretira kot logična "1" v drugem bloku. Nizko stanje na izhodu je največ 0.2V, kar ponovno zadošča za 5V CMOS vhode. Zavedati se moramo le, da smo s takšno povezavo znižali šumno mejo.

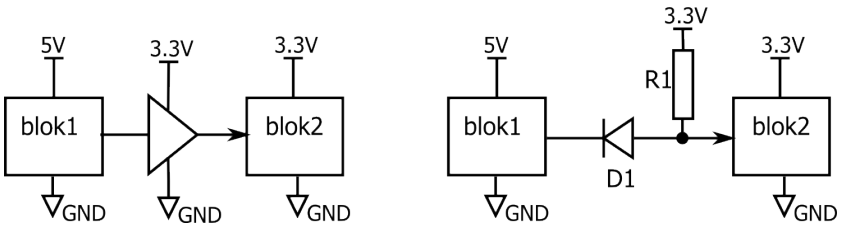


Slika 2.7: Povezovanje vezij 5V CMOS in 3.3V LVCMOS

Pri obratni povezavi iz 5V izhoda na vhod 3.3V logike moramo biti precej bolj previdni in jo brez pregleda specifikacij elementov ne smemo narediti. Visoko stanje 5V logike je namreč precej nad napajalnim nivojem LVCMOS, zato bo po signalni liniji stekel velik enosmerni tok čez vhod vezja proti 3.3V napajalni liniji. To je posledica delovanja zaščitnih diod na vhodih vezja, ki pri previsoki napetosti kratko sklenejo vhod proti napajalni povezavi. Enosmerni tok hitro uniči izhod oddajnega vezja, lahko ga pa omejimo s serijskim zaščitnim uporom (vrednosti okoli 100Ω).

Poleg toka je velik problem tudi napetost, saj sodobna vezja uniči že statična napetost na vhodu. Trend v tehnologiji integriranih vezij gre namreč proti zmanjševanju dimenzij in strukture v vezju so tako majhne, da hitro pride do preboja, ki jih trajno poškoduje. Sodobna programirljiva vezja z 3.3V priključki lahko brez poškodbe sprejmejo max. 4V napetost na vhodu. Najboljša rešitev je z uporabo namenskega ojačevalnika za pretvorbo potencialnih nivojev, za manj zahtevne primere pa naredimo vezje za omejitev napetosti iz pasivnih elementov, kot prikazuje slika 2.8.

Dioda D1 prevaja, ko je na izhodu vezja blok1 nizko stanje. V tem primeru je na diodi napetost okoli 0.7V, ki jo zazna blok2 kot logično "0" na vhodu. Kadar je na izhodu vezja blok1 napetost 5V, je dioda zaprta in na vhod vezja blok2 pride preko upora R1 napetost 3.3V. Vrednost upora R1 naj bo nekaj $k\Omega$.

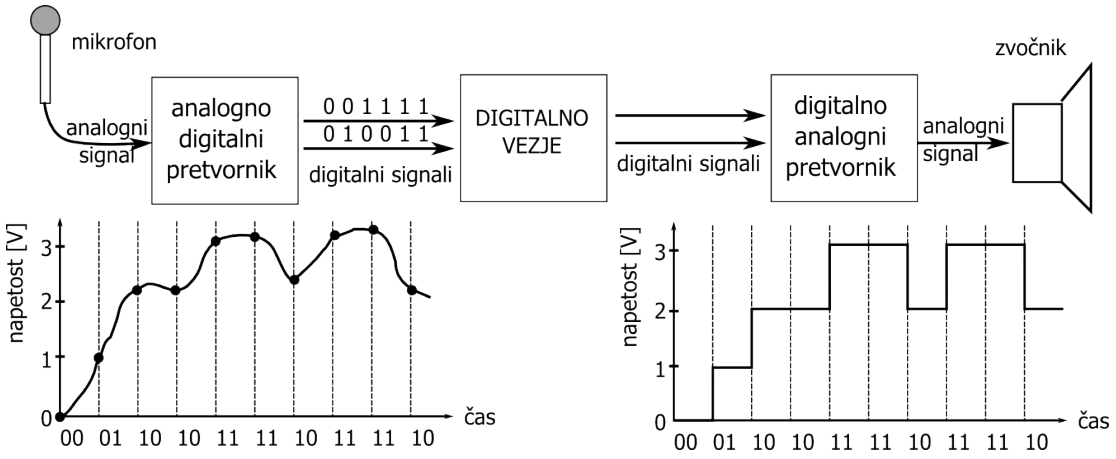


Slika 2.8: Dve možni rešitvi povezave izhoda CMOS z vhodom LVCMOS

2.3 Analogni in digitalni signali

V svetu je večina signalov po naravi analognih. Npr. avdio signal, ki predstavlja zvočno valovanje, ima neskončno možnih vrednosti jakosti (amplitude) in frekvence. Avdio signal zajamemo z mikrofonom in ga peljemo v vezje za shranjevanje in reprodukcijo. Dinamični mikrofon temelji na osnovnih principih elektromagnetizma: na membrano mikrofona, ki niha pod vplivom zvočnih valov, je pritrjen majhen magnet. Magnet je v tuljavi, v kateri se ob nihanju magneta inducira napetost. Po podobnem principu deluje zvočnik, le da v tem primeru priklopljena napetost povzroči nihanje magneta, ki s premikanjem membrane proizvaja zvok. Zvočni signal shranimo v analogni ali digitalni obliki, ki ima pred analogno več prednosti.

Analogno shranjevanje zvoka temelji na magnetnem zapisu na trak, ki pa sčasoma zgublja kvaliteto. Ob zapisovanju se na traku orientirajo skupine magnetnih delcev. Takšen zapis nima neskončne trajnosti, saj delci zaradi različnih vzrokov postopoma izgubljajo orientacijo in v reproduciranem zvoku je vse več šuma. Izgubo kvalitete lahko precej zmanjšamo, če signal pretvorimo v digitalni zapis.



Slika 2.9: Pretvorba analognega signala v digitalnega in nazaj.

Slika 2.9 predstavlja princip pretvorbe analognega signala iz mikrofona v digitalni signal, ki ga shranjuje ali obdeluje digitalno vezje. Predpostavimo, da imamo na vhodu ojačan signal iz mikrofona z napetostmi med 0 in $3V$. Pri analogno-digitalni pretvorbi bomo vrednost signala zaokrožili na najbližje celo število: $0, 1, 2$ ali $3V$. Digitalne vrednosti bomo predstavili z 2-

bitnimi dvojiškimi števili: 00 je napetost $0V$, 01 je $1V$, 10 je $2V$ in 11 je $3V$. Dvomestni zapis smo izbrali zaradi enostavnejše razlage; v praksi zapišemo zvok z vsaj 8, za avdio kvaliteto pa tudi 16 ali več biti.

Analogni-digitalni pretvornik v enakomernih časovnih intervalih vzorči analogni signal in ga pretvori v ustrezno digitalno kombinacijo, ki jo peljemo na vhod digitalnega vezja. Zahteve za shranjevanje zapisa so sedaj precej manjše: namesto poljubne amplitude moramo shraniti le vrednosti 0 ali 1. Sedaj imamo možnost zapisa na optični medij (CD), kjer shranimo vrednosti v obliki področij z močnejšim ali šibkejšim odbojem laserskega žarka. Če bomo uporabili zapis na magnetni medij (disk ali trak), bodo vrednosti še vedno shranjene v obliki različne orientacije magnetnih delcev, vendar bo precej manj občutljiv na spremembe oz. šum. Pri branju zapisa se je potrebno odločiti le ali predstavlja ničlo ali enko, ki ju lahko razločimo kljub temu da je prisoten šum. Ena izmed osnovnih prednosti digitalnih vezij je, da so odporna na šum, ki je v napravah vedno prisoten.

Digitalno-analogni pretvornik spremeni digitalne kombinacije nazaj v analogen signal, kot kaže desni diagram na sliki 2.9. Reproducirani signal ni povsem enak vhodnemu analognemu signalu, vendar se mu lahko približa s poljubno natančnostjo, če uporabimo več bitov in pogostejše vzorčenje.

Kvaliteten digitalni zapis zasede veliko prostora. Če vzorčimo signal več tisočkrat na sekundo z velikim številom bitov, pridemo do zelo velike količine podatkov. Z gradniki digitalnih vezij znamo narediti različne računske operacije in algoritme, ki omogočajo *zgoščevanje* podatkov. Zgoščevanje temelji na dejstvu, da se določene vrednosti pogosteje pojavljajo. Vzemimo preprost primer: imamo 10-bitno vzorčenje, pri katerem se pogosto pojavljata kombinaciji 0000000000 in 1111111111. Algoritem zgoščevanja zapisuje zaporedne bite. Če je prvi bit 0 in naslednji 0, naj to pomeni kombinacijo desetih ničel (0000000000), kadar je naslednji bit 1 pa kombinacijo desetih enic (1111111111). Če je prvi bit 1, pa vemo da temu bitu sledi poljubna kombinacija ničel in enic. Recimo, da imamo zaporedje ”0000000000 0000000000 0000001111 1111111111”. Po predstavljenem algoritmu ga zapišemo na kratko: ”00 00 10000001111 01”. Algoritem zgoščevanja uporabimo pred shranjevanjem ali prenosom podatkov, pred reprodukcijo pa je potrebno narediti algoritem, ki opravi inverzno operacijo. Takšne operacije brez težav naredimo z digitalnimi vezji. Najbolj znan algoritem zgoščevanja avdio signala je *mp3*, ki omogoča da na CD namesto okoli 20 zapišemo več kot 200 zgoščenih pesmi. Podobne algoritme uporabljajo tudi mobilni telefoni, ki pred prenosom zgostijo podatke in tako omogočijo večjo količino pogovorov na nekem območju.

2.4 Kodiranje digitalnih signalov

Digitalna logična vezja vsebujejo elemente, ki znajo delati le z binarnimi enobitnimi ali večbitnimi signali. Signali imajo v digitalnih sistemih zelo različne pomene: določajo stanje, predstavljajo številske vrednosti ipd. Pretvorba vrednosti iz ene oblike v drugo se imenuje *kodiranje*. V skripti bomo imenovali kodiranje (v ožjem pomenu besede) kakršnokoli pretvorbo signala iz neke oblike v binarno obliko, dekodiranje pa obratno pretvorbo, kot prikazuje slika 2.10.

Nekatere fizikalne pojave lahko opišemo z dvema stanjema: naprimer pritisnjena ali spuščena

Slika 2.10: Kodiranje in dekodiranje signalov.

tipka, noč ali dan ipd. Kodiranje takšnih pojavov je trivialno: uporabimo enobitni signal in enemu stanju pripišemo logično vrednost 0, drugemu pa 1. Če imamo več stanj, jih lahko predstavimo kot zaporedna cela števila kodirana v dvojiškem sistemu. Če vzamemo dve dvojiški števki imamo na voljo 4 kombinacije: 00, 01, 10, in 11, s tremi števkami jih dobimo 8, s štirimi 16, itn.

2.4.1 Kodiranje številskih vrednosti

Za razumevanje dvojiških vrednosti si najprej poglejmo kako so sestavljena večmestna desetiška števila. Vrednost desetiškega števila lahko zapišemo kot vsoto števk pomnoženih s koeficienti potence 10. Enice množimo z 10^0, desetice z 10^1, stotice z 10^2 itn. Primer:

$$523_{(10)} = 5 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$$

Popolnoma enako velja za dvojiška števila, le da pri izračunu desetiške vrednosti uporabimo potence števila 2. Posamezne števke dvojiškega števila pomnožimo s potencami števila 2 in seštejemo. Pri zapisu celega števila množimo skrajno desno dvojiško števko z 2^0, naslednjo z 2^1 in tako dalje. Poglejmo si primer izračuna desetiške vrednosti 4-bitnega števila:

$$1100_{(2)} = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 8 + 4 = 12_{(10)}$$

Pretvorbo najlažje naredimo tako, da nad vsako števko zapišemo ustrezno potenco števila 2 in seštejemo tiste potence, pod katerimi je binarna števka 1:

$$\begin{matrix}8&4&2&1\\1&1&0&0\end{matrix}_{(2)} = 8 + 4 = 12$$

Prvi digitalni mikroprocesor Intel 4004 je računal z 4-bitnimi vrednostmi, ki v desetiškem sistemu pokrijejo območje le ene desetiške števke. Če bi želeli računati z dvomestnimi desetiškimi števili (vrednosti med 0 in 99), bi potrebovali 7-bitni dvojiški zapis z območjem med 0 in 127. Pri delu z dvojiškimi signali, je koristno če na pamet poznamo potence števila 2, ki določajo zgornjo mejo območja vrednosti, kot prikazuje tabela 2.2.

Tabela prikazuje območja vrednosti 2- do 10-bitnih binarnih števil. Obseg vrednosti N-bitnega števila izračunamo s potenco 2^N. Območje vrednosti M-mestnega celega števila v desetiškem sistemu je 10^M. Če bi želeli ugotoviti, koliko bitov potrebujemo za določeno število mest v desetiškem sistemu, moramo rešiti neenačbo:

$$2^N \geq 10^M$$

N	N-bitna števila	2^N	območje vrednosti
2	00 - 11	4	0 - 3
3	000 - 111	8	0 - 7
4	0000 - 1111	16	0 - 15
5	00000 - 11111	32	0 - 31
6	000000 - 111111	64	0 - 63
7	0000000 - 1111111	128	0 - 127
8	00000000 - 11111111	256	0 - 255
9	000000000 - 111111111	512	0 - 511
10	0000000000 - 1111111111	1024	0 - 1023

Tabela 2.2: Območja vrednosti binarnih pozitivnih števil

$$log(2^N) \geq log(10^M)$$

$$N \cdot log(2) \geq M \cdot log(10)$$

$$N \geq M \frac{log(10)}{log(2)}$$

$$N \geq M \cdot 3,32$$

Iz enačbe hitro ugotovimo, da za zapis desetiških števil do 1.000.000 (M=6) potrebujemo kar 20-bitna števila v dvojiškem sistemu. Do sedaj smo se ukvarjali le z naravnimi števili, ki so samo pozitivna. Če želimo predstaviti cela števila, ki so pozitivna in negativna, moramo dodati še en bit za predznak.

V vgrajenih digitalnih sistemih so dolgo prevladovali 8-bitni mikroprocesorji z oznakami Intel 8051, Motorola 6000, Microchip PIC, Atmel AVR. Z razvojem tehnologije integriranih vezij danes naredijo za enako ali celo manjšo ceno 32-bitne mikroprocesorje, ki so razširjene izvedbe starejših ali pa povsem nove arhitekture (med vgrajenimi je najbolj znana ARM). Tudi 8-bitni procesorji lahko obdelujejo števila, ki jih zapišemo z več biti, in načeloma rešijo katerokoli nalogo. Vendar v tem primeru potrebujejo tudi za najenostavnejšo operacijo (npr. seštevanje ali primerjavo) več zaporednih korakov, kar se odraža v počasnejšem izvajanju programa.

V modelih digitalnih veziji moramo za zapis in obdelavo večbitnih podatkov uporabiti primerno kodiranje, saj je delo z velikimi dvojiškimi vrednostmi zelo nerodno. Poleg desetiške predstavitve se največ uporablja *šestnajstiški* zapis, ker je precej kompakten in nudi enostavno pretvorbo v dvojiškega. Vsaka šestnajstiška številka je zapisana z natanko štirimi dvojiškimi, kot prikazuje tabela 2.3. Dvojiški zapis pretvorimo v šestnajstiški tako, da združujemo in pretvarjamo po 4 številke hkrati. Primer pretvorbe 8-bitne vrednosti:

$$\begin{array}{cc} \overline{8}\,\overline{4}\,\overline{2}\,\overline{1} & \overline{8}\,\overline{4}\,\overline{2}\,\overline{1} \\ 0\,0\,0\,1 & 1\,1\,0\,0_{(2)} \end{array} = 1_{(10)} \quad 12_{(10)} = 1C_{(16)}$$

b_3	b_2	b_1	b_0	desetiško	šestnajstiško
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	10	A
1	0	1	1	11	B
1	1	0	0	12	C
1	1	0	1	13	D
1	1	1	0	14	E
1	1	1	1	15	F

Tabela 2.3: Celoštevilsko kodiranje 4-bitnih binarnih vrednosti

2.4.2 Kodiranje znakov in besedila

Digitalni sistemi, ki obdelujejo besedila, imajo na vhodu tipkovnico. Če bi vzeli računalniško tipkovnico in vsako tipko vezali na vhod digitalnega vezja, bi potrebovali več kot 50 povezav! Takšno potratno povezovanje ni potrebno, če predpostavimo, da naenkrat pritisnemo le eno tipko in pošljemo ustrezen znak v kodirani obliki. Najbolj razširjena oblika digitalnega zapisa črk, številk in drugih znakov je koda ASCII (angl. American Standard Code for Information Interchange). Koda ASCII je 7-bitna koda, ki vsebuje male in velike črke angleške abecede, številke, ločila in nekatere kontrolne znake. Slika 2.11 predstavlja znake s kodami med 32 in 126. Znak z desetiško kodo 32 je presledek!

```
 !"#$%&'()*+,-./
0123456789:;<=>?
@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]^_
`abcdefghijklmno
pqrstuvwxyz{|}~
```

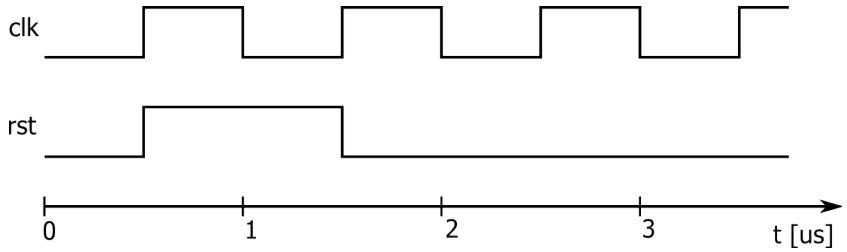
Slika 2.11: Zaporedni ASCII znaki s kodami med 32 in 126

V zapisu ASCII imajo zaporedne črke v angleški abecedi kode, ki predstavljajo zaporedne dvojiške vrednosti. Velika črka je 'A' predstavljena z binarno vrednostjo "1000001", 'B' pa z "1000010". Mala črka 'a' ima vrednost "1100001", mali 'b' pa "1100010". V podobnem zaporedju so številke od '0' do '9'. Besedo 'ABBA' zapišemo s kodami: "1000001 1000010 1000010

1000001". Zapis ASCII se uporablja v vseh napravah pri katerih zadošča za predstavitev podatkov; je zelo kompakten in zaradi svoje razširjenosti omogoča enostavno izmenjavo podatkov. V računalništvu se vse pogosteje pojavlja novejša oblika kodiranja z imenom Unicode. To je 16-bitno kodiranje, ki vsebuje bistveno večje število kombinacij in omogoča zapis latinskih, grških, arabskih, kitajskih črk in kopice posebnih simbolov.

2.5 Časovno spreminjanje signalov

Stanja digitalnih signalov, ki se spreminjajo s časom, opazujemo na časovnem diagramu (angl. waveform). Slika 2.12 prikazuje primer časovnega diagrama preprostih binarnih signalov. Običajno opazujemo več signalov, ki jih vrišemo v en diagram z enako časovno skalo za vse signale. Za opazovanje signalov, ki v vezju zelo hitro spreminjajo stanja, uporabimo ustrezen merilni inštrument: osciloskop ali logični analizator.



Slika 2.12: Časovni diagram preprostih binarnih signalov

Večbitne signale predstavimo v časovnem diagramu z zapisom njihove binarne vrednosti ali pa kar dekodirane vrednosti. Novo vrednost zapišemo ob spremembi kateregakoli bita. Slika **??** prikazuje izsek iz simulatorja, kjer za posamezen signal nastavimo način dekodiranja. Prikaz je lahko binarni, decimalni, nepredznačen decimalni, šestnajstiški, ASCII ipd.