

Osnove digitalnih tehnika i sustava u zrakoplovu (B2)

Brojevi sustavi i kodovi:

→ Općenito (baza **b**):

– Sadrži znamenke od 0 do $b-1$

– Za broj $a_{n-1} a_{n-2} \dots a_1 a_0 (b)$, gdje su a_i znamenke, a **b** je baza, vrijedi:

$$\begin{aligned} a_{n-1} a_{n-2} \dots a_1 a_0 (b) &= a_{n-1} \cdot b^{n-1} + a_{n-2} \cdot b^{n-2} + \dots + a_1 \cdot b^1 + a_0 \\ &= (((a_{n-1} \cdot b + a_{n-2}) \cdot b + a_{n-3}) \cdot b + \dots a_1) \cdot b + a_0 \end{aligned}$$

– Iz prve jednakosti slijedi postupak za pretvorbu **iz sustava s bilo kojom bazom „b“ u dekadski sustav**. Na primjer:

$$3726_{(8)} = 3 \cdot 8^3 + 7 \cdot 8^2 + 2 \cdot 8^1 + 6 \cdot 8^0 = 1536 + 448 + 16 + 6 = 2006_{(10)}$$

– Druga jednakost je dobivena izlučivanjem faktora baze gdje god je to moguće i iz nje slijedi postupak za pretvorbu **iz dekadskog u sustav s bilo kojom drugom bazom „b“**. Na primjer:

1) Podijeli početni broj sa bazom (npr. $2006_{(10)} : 8 = 250$ i ostatak 6)

2) Cjelobrojni rezultat dijeljenja zapiši ispod početnog broja

3) Ostatak pri dijeljenju zapiši desno od početnog broja

4) Ponavljaj postupak sve dok cjelobrojni rezultat dijeljenja nije 0
(dakle kada je broj koji dijeliš manji od baze)

5) Početni broj u bazi **b** bit će ostaci dijeljenja zapisani od zadnjeg prema prvom ($3726_{(8)}$ za primjer desno)

Primjer:

2006	6
250	2
31	7
3	3
0	

→ Primjeri (često korišteni sustavi):

– **Binarni:**

- baza 2; znamenke: **0, 1**
- česta oznaka (u programiranju) je 0b, dakle ako „broj“ počinje s „0b“ onda to znači da je zapisan binarno (npr. $0b101 = 5_{(10)}$)

– **Oktalni:**

- baza 8; znamenke: **0, 1, 2, 3, 4, 5, 6, 7**
- česte oznake (u programiranju) su 0 ili 0c, dakle ako „broj“ počinje s „0“ onda to može značiti da nije zapisan u dekadskom nego u oktalnom sustavu (npr. $034 = 28_{(10)}$)
- lako se pretvara u i iz binarnog sustava grupiranjem **3 po 3 znamenke** (jer je $8 = 2^3$), na primjer $101\ 001\ 110_{(2)} = 516_{(8)}$ jer je $101_{(2)} = 5$, $001_{(2)} = 1$ i $110_{(2)} = 6$

– **Heksadecimalni (heksadekadski):**

- baza 16; znamenke: **0, ..., 9, A (=10), B (=11), C (=12), D (=13), E (=14), F (=15)**
- česta oznaka (u programiranju) je 0x, dakle ako „broj“ počinje s „0x“ onda to znači da je zapisan u heksadekadskom sustavu (npr. 0x34 = 52₍₁₀₎)
- lako se pretvara u i iz binarnog sustava grupiranjem **4 po 4 znamenke** (jer je 16 = 2⁴), na primjer 1 0100 1110₍₂₎ = 14E₍₁₆₎ jer je (000)1₍₂₎ = 1, 0100₍₂₎ = 4 i 1110₍₂₎ = 14 (=E)
- često se koristi za zapis bajtova (1 bajt = 8 bitova) kao parova hekza znamenki, na primjer kod boje zadane preko RGB vrijednosti (npr. #00AAFF) ili kod zapisa MAC adrese (npr. D0:50:99:53:D3:7E)

→ Kodovi:

– Kodiranje općenito: zapis nečega na neki (drugi) način

- Zašto? Jer je korisno za neku namjenu, na primjer:

→ Skrivanje informacije (kriptiranje), npr. Cezarov kôd, Vigenеровi kodovi

→ Skraćivanje zapisa (kompresija), npr. tzv. „zipanje“

→ Prijenos podataka, detekcija i ispravljanje pogrešaka u prijenosu, ...

– U malo užem (računalnom) smislu: neki način zapisa brojeva ili znakova pomoću nula i jedinica (bitova) koji nije onaj matematički, ali je prikladan i/ili efikasan za neku uporabu.

– Primjeri kodiranja brojeva u digitalnim sustavima:

- **BCD (Binary Coded Decimals) kôd** - zapis dekadskog broja kodiranjem svake znamenke sa 4 bita

→ To nije isto što i binarni broj, iako mu sliči! Zato je važno znati kojim postupkom (kodom) su informacije zapisane da ih se ne bi krivo pročitao/interpretiralo!

→ Određena količina redundancije, odnosno neiskorištenosti (jer sa 4 bita postoji 15 kombinacija, a mi koristimo samo 10, dakle 66%), ali to je cijena koju se plaća za jednostavnost uporabe (može se čitati 4 po 4 bita, neovisno o duljini broja)

→ Primjer: 1659 → 0001|0110|0101|1001 ≠ 0001011001011001₍₂₎ = 5721₍₁₀₎

- **Grayev kôd** – susjedni brojevi razlikuju se u samo jednom bitu

→ Korisno za detekciju pogrešaka u nekim prijenosima podataka i za mnoge druge specijalizirane primjene u računalstvu

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	-
1011	-
1100	-
1101	-
1110	-
1111	-

→ Koga zanima više: https://en.wikipedia.org/wiki/Gray_code

– Primjeri kodova za otkrivanje (i ispravljanje) pogrešaka u prijenosu:

- **Paritetni bit** – dodavanje nizu bitova jedan dodatan bit koji je 1 ako niz sadrži neparan broj jedinica, a 0 ako niz sadrži paran broj jedinica; na primjer:

Niz bitova	Broj jedinica	Paritetni bit	Kodirani niz
1000101	3	1	1 1000101
110110100101010	8	0	0 110110100101010
0111011	5	1	1 0111011
01001	2	0	0 01001

→ Najjednostavniji „kôd“ za otkrivanje pogreške u prijenosu, omogućava da primatelj otkrije da se dogodila jedna (odnosno neparan broj) pogreška u jednom nizu bitova jer paritetni bit neće odgovarati stvarnom stanju

→ Ako su se dogodile dvije (ili parni broj) pogrešaka onda nije od koristi, zato se samostalno koristi samo tamo gdje je očekivani broj pogrešaka VRLO mali (npr. komunikacija unutar istog računala)

→ Tipično se dodaje nizu bitova duljine $(8 \cdot N - 1)$ kako bi kodirani niz bio dugačak „okruglo“ N bajtova

- **Hammingovi kodovi** – familija kodova za detekciju i ispravljanje pogrešaka u prijenosu

→ Više paritetnih bitova dodaje se u niz bitova na točno određen način što omogućuje otkrivanje ne samo da se dogodila pogreška nego i gdje u nizu se dogodila (pa se može ispraviti)

→ Koga zanima više: https://en.wikipedia.org/wiki/Hamming_code

– Primjeri kodiranja znakova u digitalnim računalima:

- **ASCII** – prvotno kodiranje za osnovne (engleske) alfanumeričke i posebne znakove

→ 7 bitova za jedan znak → $2^7 = 128$ znakova

→ Od tih 128 znakova, prva 32 i zadnji su tzv. neprintabilni, odnosno kontrolni znakovi. Oni u pravilu nemaju znakovni prikaz za korisnika nego služe za upravljanje u nekom smislu (ovisno o aplikaciji koja „čita“)

- npr. znak pod brojem 10 (00001010) zove se „line feed“¹ i često se (u programiranju) zapisuje kao „\n“², a znak pod brojem 13 (00001101) zove se „carriage return“³ i zapisuje se (u programiranju) kao „\r“

¹ Naziv „line feed“ dolazi od printera i označavao je naredbu printeru da pomakne papir za jedan redak

² U programiranju kosa crta unazad (*backslash* - \) koristi se za označavanje „nevidljivih“ znakova

³ Naziv „carriage return“ također dolazi do printera i označavao je naredbu printeru da pomakne glavu za pisanje na početak papira

- na Windows sustavima prelazak u novi redak (tzv. „enter“) zapisuje se kao kombinacija ta 2 znaka, znači kao „\r\n“⁴, dok se u UNIX (Linux, Mac) sustavima „enter“ zapisuje samo kao „\n“. Zato se tekstualna datoteka napravljena na npr. Linux sustavu ponekad (ovisno o programu i opcijama) neće ispravno prikazati na Windows sustavu, tj. cijela će se prikazati u jednom retku, bez „entera“.
- Ako se ASCII-kodiranom znaku doda paritetni bit taman se dobije jedan bajt po znaku (nije slučajno)
- Osnovni (originalni) kôd ne sadrži ne-engleske znakove (poput š, č, ...)
- Kasnije su napravljeni mnogi kodovi koji su zadržali iste kodove za prvih 128 znakova, ali su osmi (i eventualne dodatne) bitove iskoristili za proširenje na razne posebne i dodatne znakove (umjesto za paritet). Najvažniji primjer toga je **UTF-8**.
- **Windows-1250** – kodiranje znakova za Windows operacijski sustav koje podržava „naše“ znakove (uglavnom korišteno do Windows XP i dolaska UTF-8)
- **UTF-8** – univerzalno kodiranje za sve znakove na svijetu (moderni standard)
 - Kodiranje varijabilne duljine: neki znakovi zapisani su s jednim bajtom, a neki sa 2, 3 ili 4 bajta
 - Ideja: svi znakovi iz ASCII kodiranja zadržavaju isti kôd u UTF-8, a neke od nekorisćenih 8-bitnih kombinacija koriste se kao „naputak“ da je kôd znaka zapravo sadržan u sljedećem bajtu podataka
 - Na taj način postignuta je kompatibilnost s ASCII kodiranjem i najčešći znakovi su sadržani u samo jednom bajtu, a samo za posebne (rjeđe) znakove potrebno je više od jednog bajta
 - Npr. slovo „C“ kodirano je isto kao i u ASCII kodu, sa $01000011 = 43_{(16)} = 67_{(10)}$, ali slovo „Č“ je kodirano sa dva bajta, kao $C4\ 8C_{(16)}$, pri čemu bajt „C4“ (koji u ASCII ne postoji) signalizira računalu da treba pročitati još jedan bajt kako bi „shvatilo“ o kojem znaku se zapravo radi

⁴ Iz prethodne dvije fusnote može se zaključiti da bi ta kombinacija otprilike značila „prijeđi u početak sljedećeg retka“

Vrste i pretvorbe signala:

→ Analogni signal – neka vrijednost (napona/struje/...) u svakom trenutku

– Prednosti:

- beskonačno precizan, tj. vjeran (do na šum), i kontinuiran u vremenu

– Mane:

- obrada signala komplicirana i specifična (računala rade s digitalnim signalom)
- jako osjetljiv na šum i distorziju signala
- prijenos jednog signala zahtijeva vlastiti par žica (jer je kontinuiran u vremenu)

→ Digitalni signal – niz bitova, dakle logičkih jedinica i nula, nekako zapisanih u signalu; npr.

– jedna od 2 vrijednosti napona ($V_{\max}$ ili $V_{\min}$) koja je na žici prisutna neko vrijeme Δt

– Prednosti:

- Omogućava računalnu obradu podataka (logički sklopovi)
- Prijenos je pouzdaniji i **puno** otporniji na šum:

→ jer npr. ako je raspon 0 - 5 V onda sve vrijednosti od 2.5 do 5 V se mogu smatrati logičkom jedinicom, a sve vrijednosti od 0 do 2.5 V mogu se smatrati logičkom nulom pa je prostor za šum bez promjene informacijskog sadržaja signala ogroman (polu raspona)

- Jedna žica može prenositi podatke s više izvora (multipleksiranje)

– Mane:

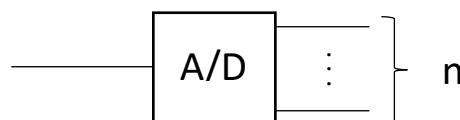
- gubi se preciznost, odnosno vjernost originalu (najbolji primjer je zvuk⁵)

→ unosi se „šum“ u odnosu na original i pri konverziji iz analognog u digitalni (gubitak informacija) i pri rekonstrukciji analognog iz digitalnog (izmišljanje izgubljenih informacija)

→ Postupak A/D pretvorbe (konverzije):

– Analogni signal (ulaz – 1 žica) očitava se svakih Δt vremena i pretvara u cijeli broj N koji se zapisuje sa n bitova (izlaz – n žica)

– To radi integrirani sklop – **A/D pretvornik** (konverter) – koji se sastoji od sklopa za očitavanje uzoraka (*sample & hold*) i sklopa za konverziju i kodiranje



⁵ https://en.wikipedia.org/wiki/High_fidelity

- Sklopovi za očitavanje uzorka (sample & hold) služe tome da zadrže iznos ulaznog signala iz nekog trenutka (uzorak) dovoljno dugo i stabilno na ulazu sklopa za konverziju i kodiranje kako bi on mogao odraditi svoj posao
- Tipovi pretvornika: paralelni (s izravnom usporedbom), brojeći (integrirajući), s postupnim približavanjem, ...

– Parametri:

- Minimalna i maksimalna vrijednost ulaznog signala: $V_{\min}$, $V_{\max}$ [fizikalni parametri]
→ **Raspon** signala = $(V_{\max} - V_{\min})$, npr. $+3 \text{ V} - (-2 \text{ V}) = 5 \text{ V}$
- Broj bitova: n [tehnološki parametar pretvornika]
→ **Razlučivost** (rezolucija) = 2^n
→ Razlučivost govori o tome na koliko se dijelova raspon ulaznog signala dijeli – ti dijelovi zovu se *pojasevi*
- Korak kvantizacije: Δ = raspon / razlučivost [izvedeni parametar procesa]
→ npr. $\Delta = (V_{\max} - V_{\min}) / 2^n = 5 \text{ V} / 2^4 = 0.3125 \text{ V}$
→ to je „širina“ jednog „pojasa“ vrijednosti kojima će biti dodijeljen isti broj

– Koraci postupka A/D pretvorbe:

- 1. Uzorkovanje:** poseban sklop na ulazu očitava ulazni analogni signal i drži ga fiksnim dovoljno vremena da se provedu ostali koraci; ta očitavanja ulaznog analognog signala zovu se *uzorci* i možemo ih označiti sa U_i (kao i-ti očitani napon)
- 2. Kvantizacija:** očitanom uzorku se zatim dodijeli redni broj pojasa u kojem se nalazi; to se može formulom:
$$N_i = \text{floor}^6((U_i - V_{\min}) / \Delta)$$
- 3. Kodiranje:** kvantizirani uzorak N pretvara se u n bitova koristeći željeni kôd, tipično se pretvara u običan binarni broj s fiksnim brojem znamenki/bitova. Zatim se cijeli postupak (sva tri koraka) ponavlja sa uzimanjem sljedećeg uzorka...

– Primjer:

- ako je raspon $-2 - 3 \text{ V}$ i preciznost 8 bitova, onda je korak $\Delta = 5/2^8 = 5/256 \approx 0.0195 \text{ V}$
- ako su redom očitane analogne vrijednosti:
 $-0.8, 0.4, 1.8, 2.7, 0.9, -1.6, \dots$
onda će redni brojevi pojasa kojima pripadaju biti (u zagradi su decimalni ostatci):
 $(-0.8 + 2)/0.0195 = 61(.54), 123(.08), 194(.87), 241(.03), 148(.72), 20(.51), \dots$
a to znači da će na izlazu iz A/D konvertera biti sljedeći niz bitova:

⁶ Funkcija „floor“ je engleski naziv za funkciju koja vraća najveći cijeli broj manji od danog broja (dakle samo skine decimalni dio, npr. $\text{floor}(4.7) = \text{floor}(4.2) = 4$)

00111101, 01111011, 11000010, 11110001, 10010100, 00010100, ...

– A/D konverzija unosi dvije vrste pogreške u signal, koje se vide kao šum u signalu nakon njegove rekonstrukcije:

- Pogreška/šum uzorkovanja (*sampling error/noise*) – dolazi od konačne vremenske razlučivosti, odnosno vremena Δt između dvaju uzimanja uzoraka
- Pogreška/šum kvantizacije (*quantisation error/noise*) – dolazi od konačne razlučivosti u iznosu signala, odnosno koraka kvantizacije Δ

→ Postupak D/A pretvorbe (konverzije):

– Očitava se n po n bitova digitalnog signala (ili po jedan sa n paralelnih vodova istovremeno ili n za redom sa jednog multipleksiranog voda) koji se pretvaraju u niz analognih uzoraka iz kojih se rekonstruira analogni signal

– To radi integrirani sklop – **D/A pretvornik** (konverter) – koji se sastoji od sklopa za dekodiranje i konverziju te rekonstrukcijskog filtera

- Rekonstrukcijski filter služi tome da izlazni signal iz sklopa za dekodiranje i konverziju (koji je stepeničast) zagladi tako da bude/izgleda/zvuči/ponaša se prirodnije; ono što rekonstrukcijski filter zapravo radi je popunjava signal izmišljajući odnosno pogađajući ono što je izgubljeno prilikom A/D konverzije

- Tipovi pretvornika: s mrežom otpornika (težinskom ili ljestvičastom), ...

– Parametri - isti kao kao A/D pretvorbe: **raspon, razlučivost i korak kvantizacije**

– Koraci postupka A/D pretvorbe:

1. Dekodiranje: n bitova sa ulaza dekodira se u cijeli broj koji predstavlja redni broj pojasa kojem taj uzorak pripada: N_i

2. Pridjeljivanje analogne vrijednosti: dekodirani broj pojasa se zatim pretvori u napon koji odgovara donjoj granici tog pojasa; to se može formulom koja je izravno matematički suprotna (inverzna) onoj u drugom koraku A/D konverzije:

$$U_i' = N_i \cdot \Delta + V_{\min}$$

3. Interpolacija (rekonstrukcija): niz napona stvorenih u drugom koraku se nekako „glatko“ povezuju pomoću rekonstrukcijskog filtera. To je zapravo „pogađanje“ signala između poznatih uzoraka koji je zapravo izgubljen prilikom postupka A/D konverzije. Pretpostavka je da je signal bio gladak, bez naglih skokova, i dovoljno gusto uzorkovan. Zatim se cijeli postupak (sva tri koraka) ponavlja sa čitanjem sljedećih n bitova...

– Primjer:

- ako je raspon -2 - 3 V i preciznost 8 bitova, onda je korak $\Delta \approx 0.0195$ V (*vidi A/D primjer*)
- ako je očitani ulazni niz:

00111101, 01111011, 11000010, 11110001, 10010100, 00010100, ...

on se prvo pretvara u brojeve pojaseva:

61, 123, 194, 241, 148, 20, ...

i zatim u izlazni niz vrijednosti:

$-2 + 61 \cdot 0.0195 = -0.8105, 0.3985, 1.783, 2.6995, 0.886, -1.61, \dots$

- ako se ove vrijednosti usporede sa početnim vrijednostima iz primjera A/D konverzije na prethodnoj stranici može se vidjeti da su *apsolutna* odstupanja ($|U_i' - U_i|$):

0.0105, 0.0015, 0.017, 0.0005, 0.014, 0.01, ...

odnosno da su *relativna* odstupanja ($|U_i' - U_i|/|U_i|$):

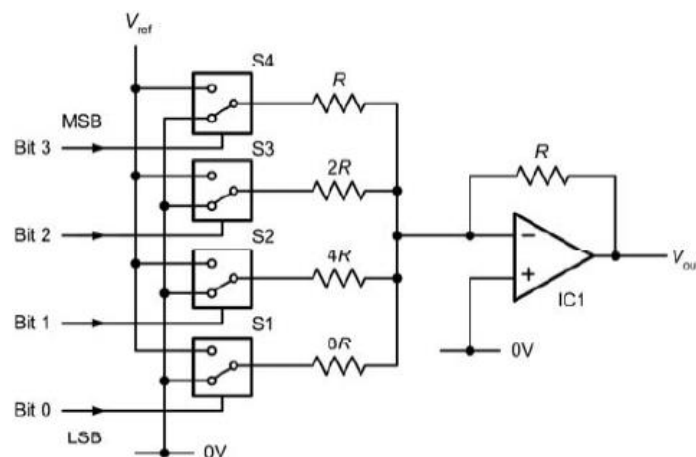
1.3%, 0.4%, 0.9%, 0.02%, 1.6%, 0.6%, ...

dakle vidimo da su odstupanja vrlo mala već sa samo 8 bitova, a tipični konverteri koriste 16, 20, 32 i više bitova...

- A/D konverzija unosi dodatnu vrstu pogreške odnosno šuma u signal koja je posljedica „pogađanja“ signala između uzoraka, odnosno rekonstrukcije signala pomoću rekonstrukcijskog filtera, tzv. **rekonstrukcijski šum** (*reconstruction noise*)

→ Princip rada nekih tipova D/A pretvornika:

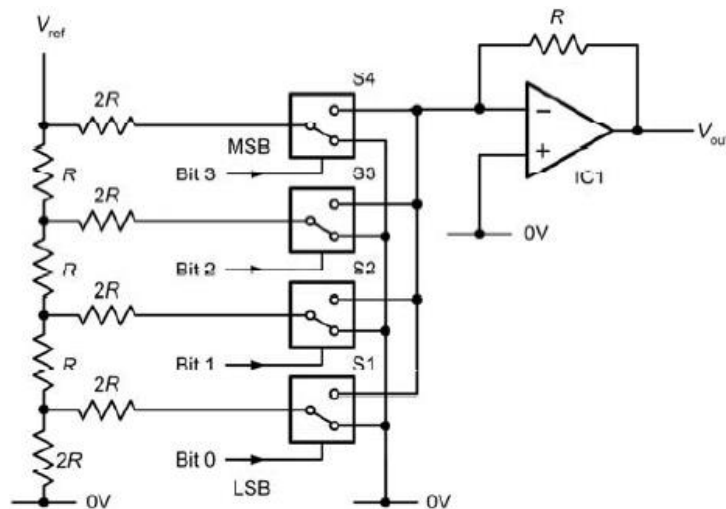
- D/A pretvornici su općenito puno jednostavniji sklopovi od A/D pretvornika, neki A/D pretvornici čak i koriste/sadrže D/A pretvornike kao dio u postupku A/D pretvorbe
- **S težinskom otpornom mrežom** (*binary-weighted*) [knjiga str. 187]



- Bazira se na principu rada zbrajala napravljenog od operacijskog pojačala
- Ulazni bitovi upravljaju poluvodičkim prekidačima (poput tranzistora, tiristora itd.) koji pojedini otpornik u mreži spajaju na visoki ili niski referentni napon
- Izlaz je napon određen otpornicima i referentnim naponima (analiza mreže i izlaza u ovisnosti o ulazima nalazi se u knjizi na str. 187)

- Prednosti: velika brzina pretvorbe, minimalnost/jednostavnost izvedbe
- Mane: mala preciznost, velika ovisnost o preciznosti/kvaliteti otpornika i referentnih napon (izvora), a to značajno utječe na njihovu cijenu
- U praksi se koriste kao maksimalno 8-bitni, obično i 4-bitni, tamo gdje je potrebna maksimalno velika brzina pretvorbe

– **S ljestvičastom otpornom mrežom (R - $2R$ resistor-ladder):**

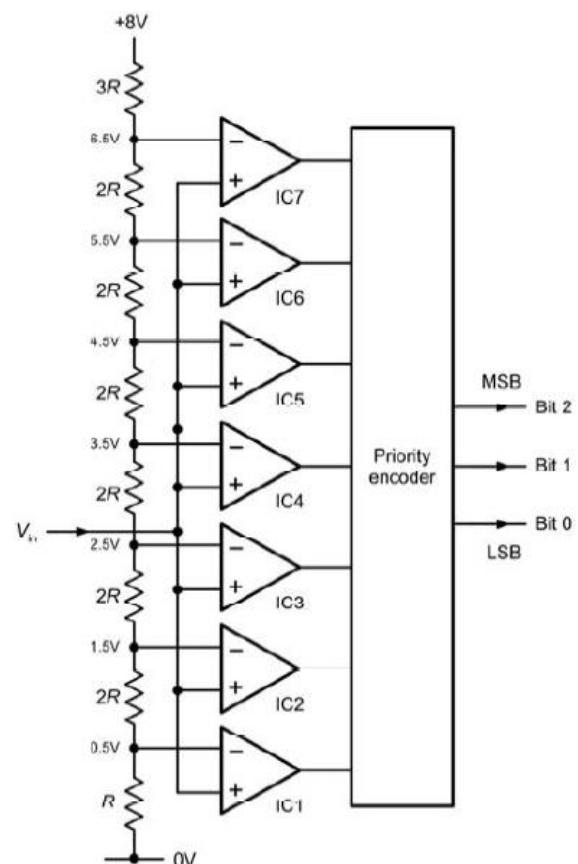


- Isti princip rada kao težinski samo drugačije spojeni otpornici drugačijih iznosa (analiza mreže i izlaza u ovisnosti o ulazima nalazi se u knjizi na str. 188)
- Prednosti: samo dvije vrste (iznosa) otpornika – puno jednostavniji i jeftiniji za integriranu izvedbu, izlaz ovisi samo o odnosu iznosa otpornika (ne i o apsolutnim iznosima) pa je zbog toga precizniji
- Mane: brzina je manja nego kod težinskog (zbog parazitskih kapaciteta)

→ Princip rada nekih tipova A/D pretvornika:

– **Paralelni** (s izravnom pretvorbom – *direct conversion/flash*):

- Bazira se na principu rada komparatora napravljenih od operacijskog pojačala – ulazni (analogni) napon (uzorak) uspoređuje se s referentnim naponima i na taj način se određuje broj pojasa kojem napon pripada (onaj

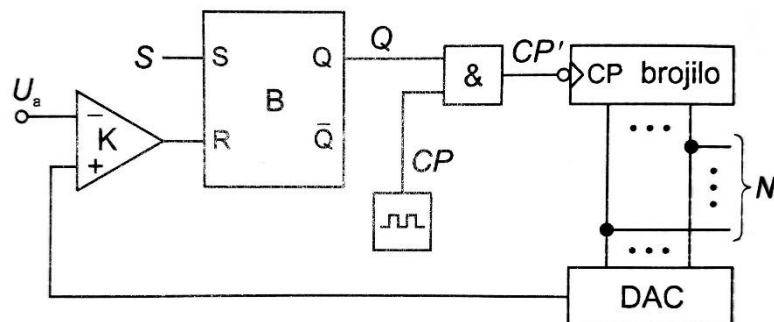


koji odgovara najvećem referentnom naponu koji je manji od uzorka), a zatim
prioritetni koder (objasni ćemo ga kasnije) kodira broj pojasa u binarni broj

- Prednosti: velika (najveća moguća) brzina, neovisna o broju bitova odnosno razina
- Mane: ogromna složenost (posljedično i cijena) za veći broj bitova - potrebno je 2^n otpornika i operacijskih pojačala (!) – zato se u praksi ne izrađuju za više od 8 bitova

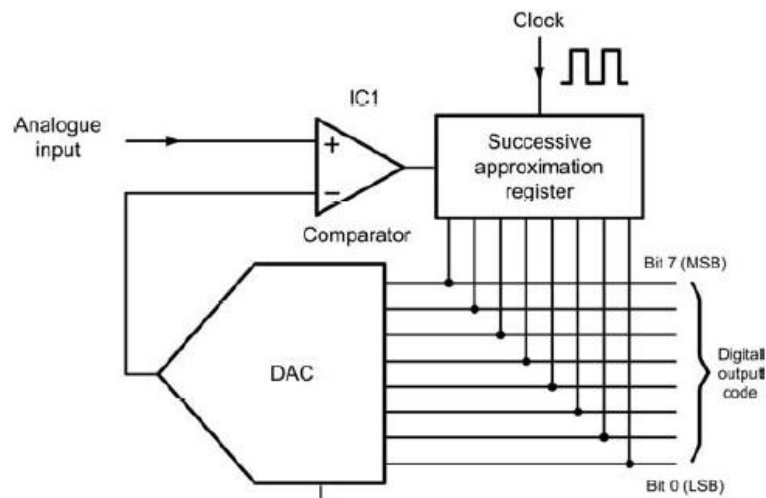
– Brojeći/integrirajući (*integrating*):

- Klasa pretvornika koji rade na principu uspoređivanja ulaznog napona (uzorka) sa naponom koji raste u vremenu – vrijeme potrebno da se naponi izjednače proporcionalno je iznosu ulaznog napona
- Brojilo broji broj taktova prije nego se naponi izjednače pa izlaz iz brojila automatski daje binarno kodirano vrijeme (koje odgovara iznosu)
- Integrirajući (s dvostrukim nagibom) A/D konverter koristi RC integrator koji generira pilasti (*ramp*) napon [*knjiga str. 192, ne morate razumijeti princip rada*]
 - Prednosti: velika preciznost i neosjetljivost na smetnje (koriste se npr. u digitalnim voltmetrima)
 - Mane: vrlo mala brzina zbog principa rada (nisu prikladni za digitalnu obradu signala, npr. audio i slično)
- Brojeći A/D konverter umjesto izvora pilastog napona koristi izlaz iz samog brojila kojeg u povratnoj vezi pretvara u napon pomoću D/A konvertera [*knjiga str. 191, trebali bi razumjeti princip rada*]



- A/D konverter koji sadrži D/A konverter
- Prednosti: relativno velika preciznost i neosjetljivost na smetnje uz relativno veliku brzinu i jednostavnost izvedbe

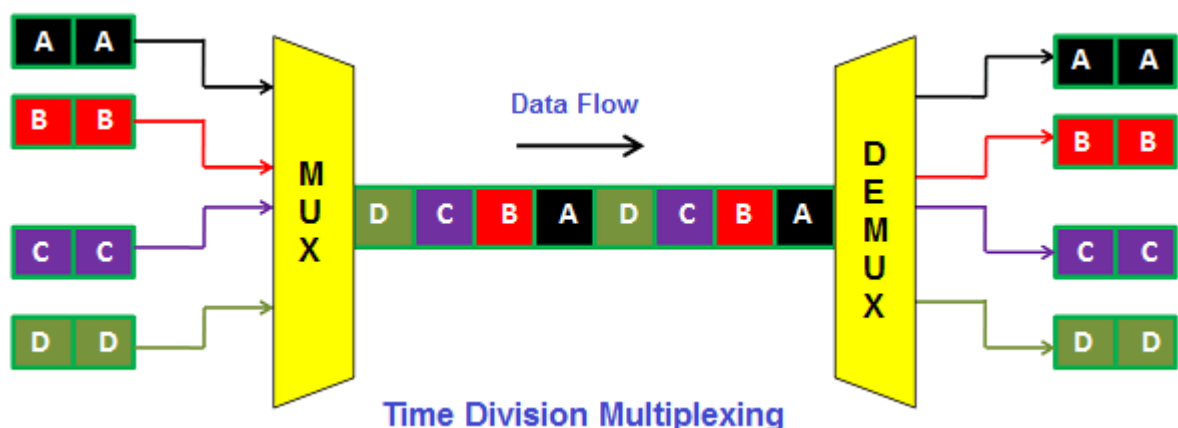
– S postupnim približavanjem (*successive approximation*):



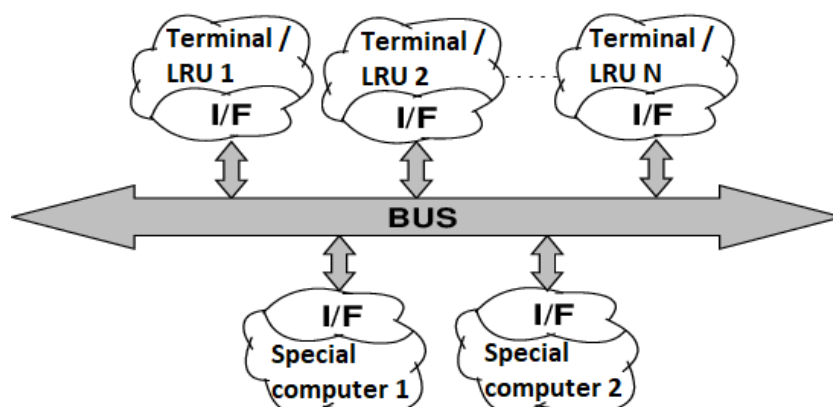
- Slična (skoro ista) struktura kao i brojeći A/D pretvornik, samo umjesto brojila ima poseban upravljački sklop koji ne broji po redu nego radi malo „pametnije“ (kako točno opisano je u knjizi na stranici 194/195; zanimljivo je, ali ne trebate to znati)
- Prednosti: veća brzina od brojećeg uz podjednaku preciznost i neosjetljivost na smetnje
- Mane: malo veća složenost od brojećeg, ali u principu zanemariva i ovakvi pretvornici se često koriste jer su dovoljno brzi, precizni i jeftini za većinu namjena

Podatkovne sabirnice:

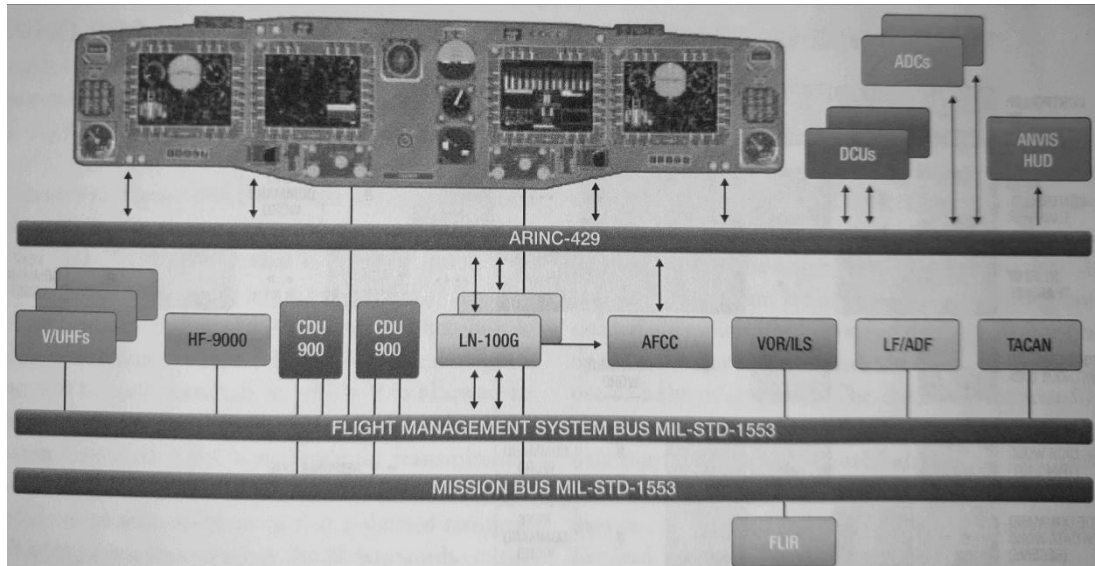
- Prijenos analognog signala: svaki uređaj/senzor treba svoj par žica → puno žica, veliki i debeli snopovi žica → teško i kvarljivo
- Prijenos digitalnog signala: mogu se prenositi paralelno (svaki na svojoj žici), ali i serijski (više signala na istoj žici)
 - **Paralelni prijenos** ima problema sa sinkronizacijom (treba moći istovremeno očitati odgovarajuće bitove sa više kanala/žica), kompliciranošću i veličinom → **zastario** (skoro uopće se ne koristi)
 - **Serijski prijenos** je jednostavniji, ali zahtijeva multipleksiranje i čini se da bi trebao biti sporiji, ali NIJE jer nema problema sa sinkronizacijom, a računala su danas jako jako brza (GHz). Danas su svi standardi serijski, npr. USB (Universal Serial Bus), Ethernet, HDMI, VGA/DVI, ...
 - **Multipleksiranje** je postupak stavljanja više signala u jedan. To se može na razne načine, a najjednostavnije je **vremensko multipleksiranje** pri čemu se N digitalnih signala stavlja u jedan tako da se puštaju naizmjenično, znači n bitova prvog signala, pa drugog, pa trećeg, i tako do zadnjeg pa onda sljedećih n bitova od prvog itd.



- **Sabirnica** (engl. *bus*) – centralni komunikacijski kanal (žica/vod/...) na koji se spajaju svi izvori i korisnici podataka (tzv. terminali ili LRU) i razna posebna računala ako su potrebna



- Može biti **višerazinska**, tj. zrakoplov može imati više sabirnica (različitih standarda čak), svaku za drugu posebnu namjenu (npr. jednu za funkcije kritične za let, a drugu za multimediju ili za naoružanje i borbene funkcije itd.), koje mogu međusobno komunicirati (na slici ispod je primjer sabirničke arhitekture S-70 helikoptera)



- Također unutar jedne logičke sabirnice mogu postojati redundantne fizičke sabirnice (žice/vodovi) za slučaj potrebe/kvara (*backup*)
- Sabirnica, odnosno (digitalni) kanal može po smjeru prijenosa biti:
- **simplex** – signal može duž jednog kanala putovati samo u jednom smjeru, dakle od izvora prema korisniku; npr. televizijski ili radio signal
 - **half-duplex** – signal može duž jednog kanala putovati u oba smjera, ali ne istovremeno; npr. *walkie-talkie* ili *speakerphone*
 - **full-duplex** – signal može duž jednog kanala putovati u oba smjera istovremeno, barem što se tiče percepcije korisnika (fizički naravno ne može jer u jednom trenutku na jednoj žici može biti samo jedan signal/napon); npr. telefon, Ethernet, ...
- Glavni fizikalni parametar sabirnice je brzina, odnosno **propusnost** (*bandwidth*) – koliko podataka može ići/proći kroz nju u jedinici vremena
- Mjeri se u bitovima po sekundi (bps), dakle treba podijeliti s 8 da se dobije brzina u bajtovima po sekundi...
 - stare modemske internetske veze su bile 55 Kbps, dakle 55 tisuća bps
 - moderne Ethernet internetske veze su do nekoliko desetaka Mbps, dakle milijuna bps
 - veze preko optičkih kablova idu i do nekoliko Gbps, dakle milijardi bps

→ Brzina i propusnost sabirnice određena je i **medijem** kroz koji signal putuje, a on u zrakoplovu može biti:

– **Žičani** – najstariji i idalje najčešći; problem je elektromagnetsko zračenje i interferencija te atenuacija (smanjenje, gubitak) signala u prijenosu; najčešći tipovi su:

- **STP** (*shielded twisted pair*) – dvije bakrene žice, isprepletene da bi se smanjilo zračenje i interferencija (tzv. bakrena parica – upotrebljava se u telefoniji i telekomunikacijama općenito) i omotane uzemljenom žičanom mrežicom koja igra ulogu Faradayevog kaveza⁷, dakle dodatno eliminira EM zračenje

→ najčešće upotrebljavana; brzine do nekoliko Mbps⁸

- **Koaksijalni kabal** (coax) – jedna deblja središnja izolirana žica (faza) omotana žičanom mrežicom (nula)

→ skuplji, krući i teži od STP

→ ima bolja svojstva, ali STP je uglavnom više nego dovoljan

→ koristi se za veće udaljenosti, posebno osjetljive signale i brzine do 50 Mbps

– **Svjetlosni (optički)** – optička vlakna prenose signal kao svjetlost, dakle EM val, a ne kao kretanje elektrona; imaju puno bolja svojstva prijenosa signala od žica i jeftina su, ali imaju i određene nedostatke i uvode određene komplikacije – više od tome kasnije

→ Zrakoplovni sabirnički standardi:

– **MIL-STD-1553B** (od 2.2018. je -1553C):

- Standard za (američke) vojne zrakoplove, svemirske letjelice, ...
- Serijska, asinkrona⁹, half-duplex sabirnica od STP kablova
- Propusnost do 1 Mbps i ne više od 1 greška na 10 milijuna bitova – **brza i pouzdana**
- Koristi 20-bitne riječi¹⁰ za naredbe i podatke
- Sastoji se od kontrolera sabirnice (Bus Controller - BC), monitora sabirnice (Bus Monitor – BM) i do 31 terminala, odnosno izvora i/ili korisnika podataka
 - BC inicira svu komunikaciju, terminali samo rade što im BC kaže (šalju i primaju)
 - BM prati status svakog odgovora i provjerava za pogreške
 - BC i BM su često fizički zajedno i čine tzv. *mission computer*

⁷ https://en.wikipedia.org/wiki/Faraday_cage

⁸ **Mbps** – *megabits per second*, dakle milijuna bitova u sekundi; isto tako **Kbps** (kilo) i **Gbps** (giga)

⁹ Asinkrona sabirnica znači da se komunikacija događa samo na zahtjev, ne stalno u pravilnim intervalima

¹⁰ U računalima „riječ“ (engl. WORD) označava logičku cjelinu s kojom procesor barata, dakle koliko bitova procesori obrađuju odjednom, odnosno koliko ih putuje niz sabirnicu u „paketima“

- Terminali se na sabirnicu spajaju preko transformatora, ne direktnim fizičkim spojem, zato da ne oštete sabirnicu u slučaju kvara, npr. kratkog spoja

– **ARINC 429:**

- Najčešće korišten standard (i dan danas) u komercijalnim i transportnim zrakoplovima
- Serijska, asinkrona, simplex sabirnica od STP kablova
 - potreban je **jedan STP kabel po izvoru podataka**, na njemu samo taj izvor šalje, a svi drugi terminali samo primaju (jer je sabirnica simplex)
- Manje kompleksna i jeftinija, ali sporija i manje pouzdana sabirnica od MIL-STD-1553
- Nema kontrolera i monitora sabirnice i prima maksimalno 20 terminala
- Propusnost do 100 Kbps
- Koristi 32-bitne riječi

– **ARINC 629:**

- Noviji od ARINC 429, napravljen za Boeing 777, koristi se još i na 737 MAX, A330 i A340
- Napravljen da bude veće propusnosti i pouzdanosti te da može imati više terminala (izvora i korisnika podataka)
- Half-duplex, kao i MIL-STD-1553, ali bez kontrolera i monitora sabirnice
 - zato treba poseban protokol i poseban hardware zbog čega nije baš postao opće prihvaćen i čest standard
- Koristi 20-bitne riječi (kao MIL-STD-1553), propusnost do 2 Mbps i max 128 terminala

– **ARINC 664 AFDX (IEEE 802.3 – Ethernet):**

- Malo prilagođeni Ethernet¹¹ standard (serijski, full-duplex, STP ili optika);
AFDX – *Avionics Full-Duplex Switched*
- Radi kao telefonska centrala – prespaja između korisnika (stvara virtualne izravne veze, kao *router* u LAN mreži doma)
- Propusnost od 100 Mbps do 1 Gbps i više

– **IEEE 1394 Firewire:**

- Standard razvijen za potrebe visoke propusnosti (800 Mbps i više) kod običnih korisnika (*streamanje* videa i slično) – tipično se koristi u svjetlovodna vlakna
- Zrakoplovna industrija preuzela za *entertainment* sustave u putničkim avionima i slično, ali korišten također i u vojne svrhe (npr. u F-35 za kontrolu leta i motora)

¹¹ Ethernet standard (službeno ime IEEE 802.3) je ono što svi imamo doma, oni kao malo deblji telefonski kablovi kojima spajamo računalo s *routerom* ili računala međusobno u LAN mreži

Svjetlovodni (optički) sustavi:

→ Opća svojstva optičkog prijenosa podataka su **veća brzina** i **veća propusnost** podataka uz **manje gubitke** i **manji šum**

- U zrakoplovima potrebno za **brzine veće od 100 Mbps** (AFDX i FireWire)
 - trenutno uglavnom samo za *in-flight entertainment* u velikim putničkim zrakoplovima
- velika tehnička prednost je **imunost na elektromagnetsku interferenciju**, dakle niti stvara zračenje koje ometa druge instrumente, niti vanjsko EM zračenje utječe na njih
 - to je načelno velika prednost, ali žičani prijenos je još uvijek uglavnom dovoljno dobar

→ Fizička svojstva sustava optičkog prijenosa signala:

- prenosi se svjetlost (EM valovi) optičkim vlaknima, umjesto naboja (elektroni) žicama
 - svjetlost koja se prenosi je crvena (650 nm) ili infracrvena (850, 1100, 1500 nm)
 - kroz jedno vlakno se **istovremeno** može prenositi **više signala na različitim valnim duljinama** (wavelength division multiplexing – WDM)
 - gubitci dolaze od dva fizikalna procesa: **apsorpcije** i **raspršenja**
 - raspršenje izobličava signal, dakle unosi šum
 - raspršenje je manje na većim valnim duljinama, zato se i koristi infracrveno
 - apsorpcija znači da materijal upija dio svjetlosti (grije se)
 - apsorpcija se događa na određenim valnim duljinama koje su izbjegavaju
- optička vlakna su prozirna vlakna sa tamnijim omotom koji reflektira svjetlost i nekoliko slojeva zaštitnog omota
 - prozirna jezgra može biti **staklena** ili **plastična**:
 - staklena je **bolja** što se tiče gubitaka, ali **skuplja** i mehanički **osjetljivija** od plastične
 - plastična vlakna mogu podnijeti veće kutove svijanja i koriste se za kraće udaljenosti – prikladna za zrakoplove
 - vlakna su jako čvrsta (duplo tanja od žica za istu čvrstoću), tanka i lagana – super za zrakoplove
 - postoje **singlemode** i **multimode** vlakna:
 - singlemode su **JAKO** tanka i prenose samo jedan signal na jednoj valnoj duljini, ali postižu **veće brzine** i **veće udaljenosti** zbog manjih gubitaka
 - multimode su deblja i mogu prenositi više signala iste valne duljine, ali imaju veće gubitke i prigušenje i postižu manje brzine – koriste se za lokalni prijenos

→ Tehnička svojstva sustava optičkog prijenosa signala:

- Potrebni su posebni **odašiljači** (engl. *transmitters*) koji pretvaraju električni signal (iz npr. senzora ili računala) u svjetlosni – sadrže **LED** ili **laser diodu** kao aktivni element
 - LED su jeftinije, ali manje snage i sporije
- Potrebni su i posebni **primatelji** (engl. *receivers*) koji pretvaraju svjetlosni signal nazad u električni – sadrže foto-diodu i slični foto-element kao aktivnu komponentu
- Također, za veće udaljenosti potrebni su i **posebni pojačivači signala**, slično kao za bežični signal, a to je nešto što običnim žicama ne treba
- Posebna tehnička teškoća i velika razlika u odnosu na obične žice je **spajanje**
 - Dva vlakna trebaju biti savršeno spojena da ne bi došlo do refleksije i gubitka signala
 - Potrebni su **posebni postupci i uređaji** za spajanje i prijenos signala među vlaknima, ne može se jednostavno ostvariti kontakt ili na brzinu popraviti kao kod običnih žica
- Prednost je što ako vlakno pukne nema signala uopće i zna se u čemu je i gdje problem, nije kao kod žica gdje postoje djelomična puknuća i loši spojevi u kojima signal „treperi“
- Postoji više vrsta spojnih i razdjelnih uređaja za optička vlakna:
 - **4 port directional** (usmjerena spojnica sa 4 kontakta) – 2 ulazna vlakna se dovedu u kontakt tako da na svakome od dva izlaza ima dio signala iz oba ulazna vlakna
 - **T coupler** (T razdjelnik) ili **Y coupler** (Y razdjelnik) – 1 ulazno vlakno na dva izlazna u željenom omjeru (signal pada linearno s brojem terminala)
 - **Star coupler** (zvjezdasta spojnica) – više ulaznih na više izlaznih vlakana odjednom
 - **Tree coupler** (stablasti razdjelnik) – 1 ulazno vlakno ravnomjerno na N izlaza
 - **WDM coupler** (WDM razdjelnik) – 1 ulazno vlakno na 2 ili više izlaznih, ali ne po snazi (količini svjetlosti) nego po valnoj duljini (dakle razdvajanje valnih duljina)
 - **Switch** (sklopka) – za razliku od ostalih, ovo je aktivni uređaj koji radi fizičko prespajanje pomoću releja ili na neki drugi način
- Također, potrebni su posebni priključci (konektori):
 - Nužno je da su vrlo precizni i savršeno ispravni, iz istih razloga kao kod spojnica
 - Postoji više standarada, za različite namjene
 - U zrakoplovima najčešće korišten je **SMA** (Sub-Miniature version A)



Slika 1 – SMA optički priključak

Digitalna logika i logički krugovi:

→ **Booleova¹² algebra** – „matematika“ za dvije vrijednosti (npr. 0 i 1)

- Ako se digitalna „0“ (niski napon) poistovjeti sa „laž“, u smislu netočne tvrdnje, a digitalna „1“ (visoki napon) poistovjeti sa „istina“, u smislu točne tvrdnje, onda Booleova algebra u dobroj mjeri odgovara logici
 - Zato se manipulacija nulama i jedinicama u računalu zove digitalna logika
- Osnovne i najjednostavnije operacije Booleove algebre su logičko „I“ (AND) koja se označava kao množenje, logičko „ILI“ (OR) koja se označava kao zbrajanje, te negacija (NOT) koja se označava sa crtom iznad izraza
 - Logičko „I“ kao rezultat daje 1 samo ako su sve ulazne vrijednosti 1, što odgovara logici da sve tvrdnje moraju biti istinite (i jedno i drugo i treće itd.)
 - Logičko „ILI“ kao rezultat daje 1 ako je barem jedna ulazna vrijednost 1 (jedna ili više njih), što odgovara logici da barem jedna tvrdnja mora biti istinita (jedno ili drugo ili treće itd., ali može biti i više njih istovremeno, nije *isključivo-ili*)
 - Logičko „NE“, odnosno negacija daje rezultat suprotan od ulaza, što odgovara logici da je negacija od laži istina i obrnuto
 - Sve druge logičke operacije i uopće sve moguće funkcije Booleove algebre mogu se izvesti/zapisati koristeći samo te tri operacije
 - Takav skup operacija (pomoću kojih se mogu izvesti/zapisati sve ostale) zove se **potpuni skup operacija**
 - Osim tog I-ILI-NE potpunog skupa operacija, koji je ljudima najprirodniji, najjednostavniji i najlogičniji, potpun skup logičkih operacija može činiti i samo jedna operacija, npr. NI (NAND) ili NILI (NOR) – više o tome kasnije
- Booleova algebra sadrži pravila (vidi tablicu ispod) pomoću kojih je moguće preoblikovati izraze, minimizirati ih (smanjiti broj operacija) i općenito manipulirati njima kao i u „normalnoj“ matematici

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$

¹² https://en.wikipedia.org/wiki/George_Boole

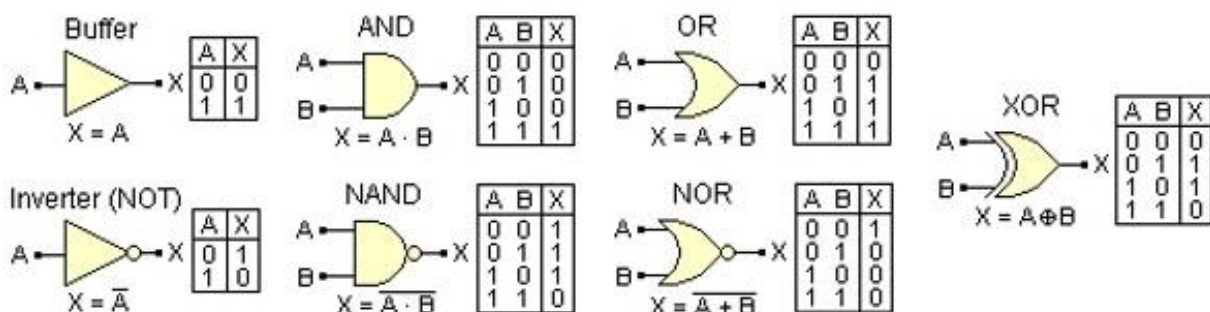
- Većina gore navedenih pravila ista je kao i u „normalnoj“ matematici s brojevima, logična su i očita, ako se malo o njima razmisli
- Glavna razlika između pravila Booleove algebre i uobičajene matematike je simetrija između „množenja“ (operacije „I“) i „zbrajanja“ (operacije „II“) koja se vidi u zakonu distributivnosti i koja je vrlo neobična zbog naviknutosti na pravila uobičajene matematike s brojevima (pa to treba posebno imati na umu)
- Uz tu razliku ključna razlika su **De Morganovi zakoni** koji povezuju sve tri osnovne operacije i koji su iznimno važni za manipuliranje izrazima Booleove algebre te ih obavezno treba zapamtiti i stalno imati na umu (ja ih pamtim kao – *cijepanje negacije mijenja operaciju*)

→ Logičke operacije/funkcije, odnosno njihovo djelovanje s obzirom na ulazne vrijednosti zapisuje se pomoću **tablica istinitosti**

- To su tablice koje sadržavaju izlaznu vrijednost funkcije/operacije (0 ili 1) za sve moguće kombinacije ulaznih vrijednosti
- Budući da vrijednost svakog ulaza (varijable) može biti samo 0 ili 1, dakle dvije su mogućnosti, onda je broj svih mogućih kombinacija ulaznih vrijednosti (dakle redaka tablice) jednak 2^N , gdje je **N** broj ulaza, odnosno varijabli te funkcije koju zapisujemo
- Tablice istinitosti za osnovne logičke operacije dvije varijable nalaze se na slici ispod
- Dodatna zanimljivost Booleove algebre, odnosno činjenice da postoje samo dvije moguće vrijednosti, je i ta da postoji konačan broj uopće mogućih funkcija N varijabli, a to je $2^{(2^N)}$, npr. postoji samo 16 logičkih funkcija 2 varijable (razmislite zašto – ima veze s brojem mogućih različitih tablica istinitosti), ali nisu nam sve jednako korisne i važne

Primjer:

A	B	C	$A \cdot \bar{B} + C$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1



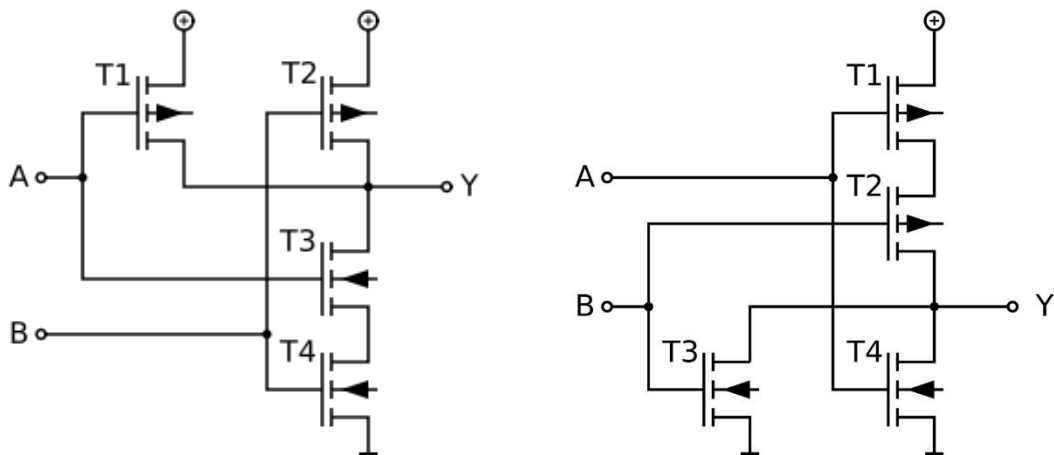
→ **Logička vrata** – elektronički sklop sastavljen od tranzistora koji obavlja logičku operaciju nad ulaznim digitalnim signalima, odnosno kraće - *elektronička implementacija logičke operacije*

- Svaka logičkih vrata (koja odgovaraju nekoj Booleovoj funkciji) imaju svoj znak i uz njih se veže tablica istinitosti i Booleov izraz koji provode (vidi sliku iznad)
 - Općenito kružić označava negaciju i može se nacrtati bilo na ulazima bilo na izlazima bilo kojeg standardnog simbola logičkih vrata (AND, OR, XOR)
 - Logička vrata I (AND) i ILI (OR) s **negiranim izlazom** imaju posebne nazive: **NI** (NAND) i **NILI** (NOR). To su u praksi vrlo česta logička vrata (najčešća zapravo).
 - Logička vrata I (AND) i ILI (OR) vrata s **negiranim ulazima** također imaju posebne nazive: **NEG-I** (NEG-AND) i **NEG-ILI** (NEG-OR). Nazivi su vrlo slični i zato treba biti jako oprezan da ne dođe do zabune.
 - Primjernom De Morganovih pravila lako se dokazuje da je NEG-I isto što i NILI, a NEG-ILI isto što i NI (provjerite to)
 - Također, simboli logičkih vrata za I i ILI mogu se crtati s **proizvoljnim brojem ulaza**, ali za isključivo-ILI se to u pravilu ne radi jer rezultat nije intuitivan
- **Pozitivna logika** – digitalna „1“ (logička „istina“) je $U_{\max}$ (visoka razina napona), a digitalna „0“ (logička „laž“) je $U_{\min}$ (niska razina napona)
- **Negativna logika** – suprotno od pozitivne, dakle „1“ je niska razina napona, a „0“ visoka razina napona
- Tehnologije logičkih vrata:
 - TTL (*transistor-transistor logic*)
 - bipolarni tranzistori, rade na „struju“
 - pozitivna logika na točno +5 V (ispravan rad vrlo osjetljiv na odstupanja)
 - troše dosta energije, imaju problem s grananjem¹³, ali su **robusti**
 - prva tehnologija masovne proizvodnje, prvog naglog razvoja i proboja digitalne elektronike – danas zastarjela
 - ne koriste se više za veće sklopove samo za pojedinačna vrata i vrlo jednostavne logičke krugove (jer su **robusti**, dakle otporni na velike struje, napone, izboje, temperature itd.)
 - CMOS (*complementary*¹⁴ MOSFET)
 - unipolarni MOSFET tranzistori (kombinirano P i N tipa), rade na „napon“

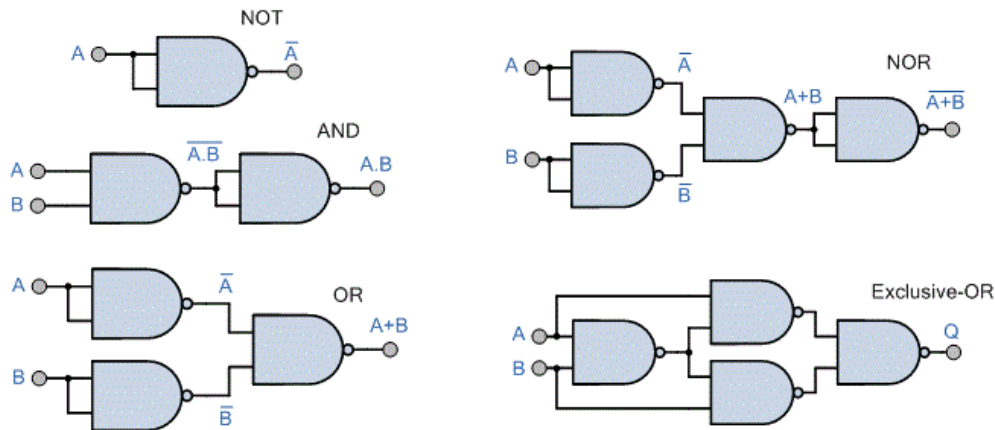
¹³ Grananje je spajanje izlaza iz jednih na ulaze više logičkih vrata – budući da bipolarni tranzistori rade na „struju“ onda je grananje problem jer se struja dijeli (1. Kirchhoffov zakon)

¹⁴ *Complementary* znači komplementarni (međusobno nadopunjujući), a dolazi od činjenice da se u istim sklopovima koriste i n- i p-kanalni tranzistori, koji se i u proizvodnom procesu i u funkcioniranju efikasno nadopunjuju.

- pozitivna logika na većem rasponu napona (+3 do +18 V)
- troše puno manje energije i puno manje se zagrijevaju, ali su **vrlo osjetljivi** (potreban je oprez pri rukovanju – više o tome u predzadnjem poglavlju o elektrostatskoj osjetljivosti)
- minijaturizacija - danas već i preko 100 milijuna tranzistora po mm²
- to je tehnologija kojom se danas izrađuje skoro sva digitalna elektronika, dakle svi čipovi, mikroprocesori itd. – **moderni standard**
 - zbog toga MOSFET tranzistori su najproizvođeniji „uređaji“ u povijesti čovječanstva ($> 10^{22}$ komada)
- U praksi se za izradu modernih integriranih sklopova (procesora, memorija, svega što je u vašim računalima, mobitelima itd.) koristi *samo jedan tip logičkih vrata* – NI (češće) ili NILI (rjeđe). Razlog tome je što to dvoje logičkih vrata svaka za sebe čine potpuni skup operacija, a istovremeno savršeno odgovaraju CMOS tehnologiji izrade u kojoj se izrađuju maksimalno jednostavno i efikasno od minimalne količine tranzistora – njih samo 4 (vidi slike ispod) – i bez ikakvih dodatnih elemenata.



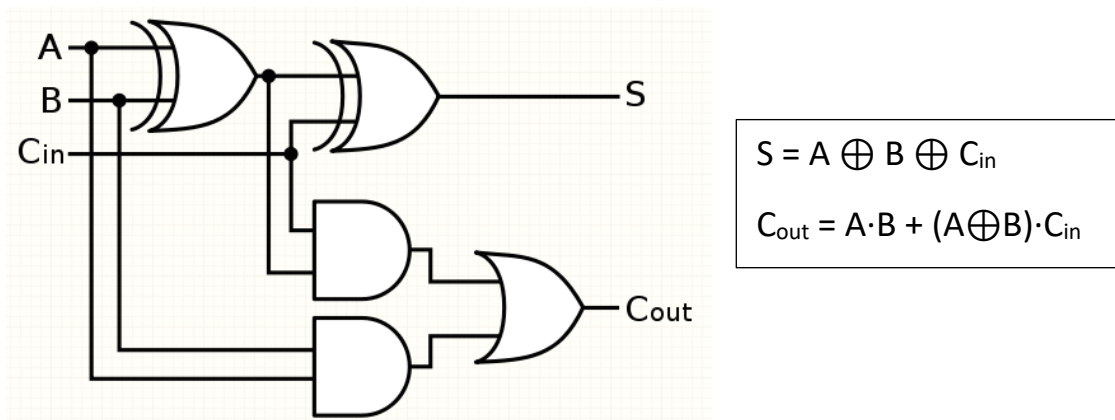
- Da NI (NAND) i NILI (NOR) čine potpune skupove operacija može se dokazati pravilima Booleove algebre ili tablicama istinitosti. Na slici ispod prikazane su kombinacije NI vrata kojima su izvedene sve druge osnovne logičke operacije (što znači da mogu biti i sve druge logičke funkcije). Zapišite Booleove izraze koji se dobiju izravno iz skica pa primjenama pravila Booleove algebre i tablicama istinitosti dokažite da su konačne operacije zaista te koje na slici piše da jesu.



- Isključivo-ILI (XOR) jako su važna logička vrata jer su ključna za izvedbu sklopova za izvođenje aritmetike s binarnim brojevima (zbrajala). To je razlog zašto se češće kao univerzalna logička vrata koriste NI, a ne NILI, jer je za izvedbu jednih EX-ILI vrata potrebno 4 NI vrata (vidi sliku iznad), odnosno 5 NILI vrata. Ta razlika od jednih vrata čini se mala i nevažna, ali ako malo više razmislite shvatit ćete da je to povećanje od 25%.

→ **Logički krugovi** (engl. logic circuits) – to su mreže logičkih vrata gdje je apstrahirana izvedba i tehnologija, dakle nije važno kakvi i koji i koliko tranzistora, koliko su i kakve i gdje struje i naponi itd., nego se pretpostavlja idealan digitalni signal i koriste se samo simboli logičkih vrata i pravila Booleove algebre za analizu rada sklopa (dakle koju logičku funkciju sklop obavlja nad ulazima)

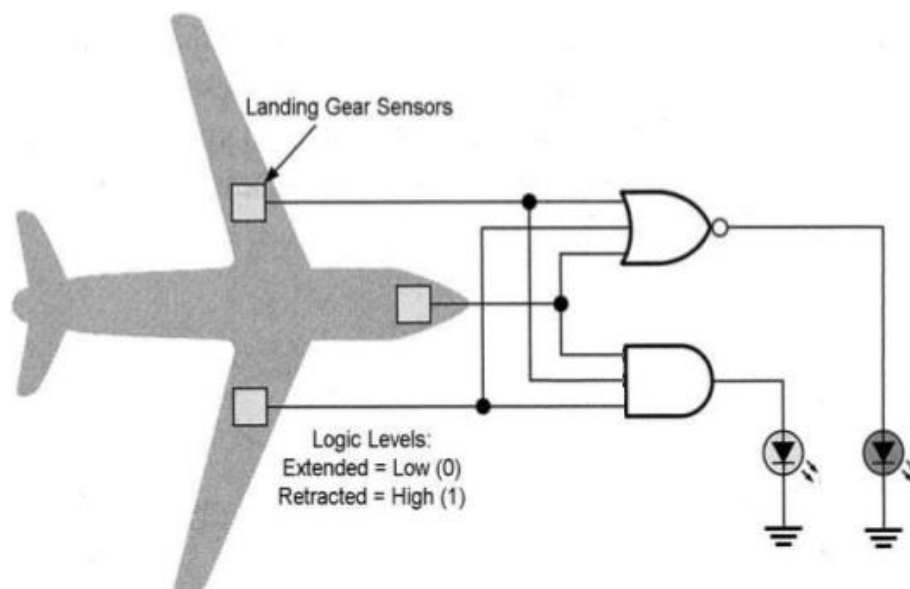
- Tipičan jednostavan primjer toga je zbrajalo koje uzima tri jednoznamenakasta binarna broja (3 bita) – 2 od pribrojnika (A i B) i jedan bit „prijenosa“ (C od „Carry“) – i vraća njihov (dvobitni) zbroj.



→ Za vježbu napišite tablicu istinitosti za sklop zbrajala (*full-adder*) koji je dan na slici iznad i uvjerite se da zaista kao rezultat daje aritmetički zbroj ulaznih bitova; rezultate možete provjeriti Googleanjem (npr. „full adder truth table“)

– Jednostavan primjer logičkog kruga u zrakoplovu bio bi npr. sklop za indicaciju spuštenosti ili uvučenosti sva tri stajna trapa zrakoplova...

- Ulazi u taj krug/sklop bi bili izlazi iz nekih senzora kod stajnih trapova koji bi indicirali da je stajni trap spušten ili uvučen; uzmimo da senzori daju nizak napon (logička „0“) za spušten stajni trap, a visok napon (logička „1“) za uvučen stajni trap
- Takav sklop bi onda trebao imati 2 izlaza: jedan koji bi bio aktivan (logička „1“) kada su sva tri stajna trapa spuštена, a neaktivan (logička „0“) inače, i drugi koji bi bio aktivan kada su sva tri stajna trapa uvučena, a neaktivan inače
- Možemo zamisliti da je prvi od tih izlaza spojen izravno na crvenu LED lampicu na nekom panelu u kokpitu, a drugi isto tako na zelenu LED lampicu pokraj. To bi onda značilo da ako su sva tri stajna trapa spuštена da gori crvena lampica, ako su sva tri stajna trapa uvučena da gori zelena lampica, a ako nije niti jedno niti drugo stanje onda ne gori nijedna lampica (što pilotu govori da je proces spuštanja/uvlačenja u trapova u tijeku ili da se nešto zaglavilo)
- Prvi izlaz, dakle sva tri stajna trapa spuštена, mora biti logička „1“ samo ako su izlazi sa sva tri senzora logička „0“, dakle mora vrijediti da nije aktivan **niti** senzor 1 **niti** senzor 2 **niti** senzor 3, a to je suprotno od operacije logičko ILI (koja daje „1“ za sve kombinacije ulaza OSIM te) što je dakle operacija **logičko NILI** između ta tri ulaza... *Napišite tablicu istinitosti!*
- Drugi izlaz, dakle sva tri stajna trapa uvučena, mora biti logički „1“ samo ako su izlazi sa sva tri senzora također logička „1“, dakle mora vrijediti da je aktivan **i** senzor 1 **i** senzor 2 **i** senzor 3, što znači da je potrebna operacija obično **logičko I** između ta tri ulaza... *Napišite tablicu istinitosti!*



→ Općeniti zapis i minimizacija logičkih funkcija

- Postoje 2 načina za sistematično, izravno, ali istovremeno i vrlo ne-minimalno, zapisivanje bilo koje logičke funkcije iz njene tablice istinitosti (dakle željenih izlaza za poznate ulaze); to su preko **mintermi** i preko **makstermi**
- **Minterm** je logička operacija nad ulaznim varijablama koja daje „1“ SAMO za jednu kombinaciju ulaznih vrijednosti i „0“ za sve ostale kombinacije, a to je svojstvo operacije **logičko I** koja daje „1“ samo kad su SVE ulazne vrijednosti „1“
 - Na primjer ako želimo da je izlaz funkcije $m = 1$ samo za ulaznu kombinaciju $A = 0$, $B = 1$ i $C = 0$, onda to znači da mora istovremeno biti i $A = 0$, odnosno $\bar{A} = 1$, i $B = 1$ i $C = 0$, odnosno $\bar{C} = 1$, što sve skupa daje $m = \bar{A} \cdot B \cdot \bar{C}$
 - Ono kako cijelu, proizvoljnu funkciju prikazujemo preko mintermi je da napišemo minterm za svaki redak iz tablice te funkcije za koji je izlaz funkcije „1“ i onda napravimo **logičko ILI** između svih tih mintermi (jer će izlaz biti „1“ kada je „1“ ili za prvu kombinaciju ulaza ili za drugu kombinaciju ulaza ili za treću kombinaciju ulaza itd.)
 - Dodatna objašnjenja i primjere pronađite u knjizi na stranicama 47-49
 - Ovaj način zapisa funkcije koristi se kada je broj kombinacija ulaza (redaka) za koji je izlaz funkcije „1“ veći od onih za koje je izlaz funkcije „0“ (jer je tada kraći nego zapis preko maxtermi)
- **Maxterm** je logička operacija nad ulaznim varijablama koja daje „0“ SAMO za jednu kombinaciju ulaznih vrijednosti i „1“ za sve ostale kombinacije, a to je svojstvo operacije **logičko ILI** koja daje „0“ samo kada su SVE ulazne vrijednosti „0“
 - Na primjer ako želimo da je izlaz funkcije $M = 0$ samo za ulaznu kombinaciju $A = 0$, $B = 1$ i $C = 0$, onda to znači da mora istovremeno biti i $A = 0$ i $B = 1$, odnosno $\bar{B} = 0$, i $C = 0$ što sve skupa daje $M = A + \bar{B} + C$
 - Ono kako cijelu, proizvoljnu funkciju prikazujemo preko maxtermi je da napišemo maxterm za svaki redak iz tablice te funkcije za koji je izlaz funkcije „0“ i onda napravimo **logičko I** između svih tih maxtermi (jer će izlaz biti „1“ samo kada nije „0“ za prvu kombinaciju ulaza i za drugu kombinaciju ulaza i za treću kombinaciju ulaza itd.)
 - Dodatna objašnjenja i primjere pronađite u knjizi na stranicama 49-50
 - Ovaj način zapisa funkcije koristi se kada je broj kombinacija ulaza (redaka) za koji je izlaz funkcije „0“ veći od onih za koje je izlaz funkcije „1“ (jer je tada kraći nego zapis preko mintermi, dakle upravo suprotno)
- Funkcija se preko mintermi ili maxtermi lako i jednostavno može točno zapisati, ali taj zapis je daleko od minimalnog, što znači da koristi puno više logičkih operacija nego što

je nužno potrebno (a to je problem prilikom fizičke implementacije pomoću logičkih vrata)

- Zato se logičke funkcije/operacije/sklopovi prije implementacije **minimiziraju**, što se treba raditi s obzirom na skup operacija s kojima se radi (dakle logičkih vrata koja su nam dostupna za implementaciju sklopa)
- Postoji nekoliko načina minimizacije funkcija:
 - 1) Ručna minimizacija/sređivanje/svođenje na druge operacije izravnom primjenom pravila Booleove algebre
 - primjenjuje se kod vrlo jednostavnih funkcija
 - 2) Ručna minimizacija primjenom K-tablica¹⁵ – svodi samo na skup operacija I, ILI, NE
 - može koristiti i kod kompliciranijih funkcija više varijabli, ali u praksi se ne koristi
 - 3) Računalna minimizacija primjenom raznih algoritama i specijaliziranih programa
 - Za jednostavne, ručno napisane funkcije/sklopove postoje jednostavni programi koji implementiraju više-manje ideju K-tablica (čisto da se ne mora to raditi ručno)
 - Za kompliciranije sustave minimizacija se automatski provodi unutar specijaliziranih sustava/programa za dizajniranje digitalnih sklopova

→ **Sveobuhvatni primjer: 7-segmentni prikaz broja (TODO)**

- Tablica sa BCD ulazima i 7-segmentnim izlazima (stranica 147 u knjizi)
- Svaki od segmenata (a-g) je logička funkcija koja se može zapisati preko mintermi/maxtermi i onda minimizirati...
- Takva implementacija bila bi minimalna u smislu broja korištenih operacija odnosno logičkih vrata, ali postupak je mukotrpan i gnjavaža i konačni sklop izgleda komplicirano. Puno jednostavnije je to napraviti pomoću gotovih generičkih sklopova koji nisu minimalni, ali imaju jednostavnu strukturu – koder i dekodeer.
 - Vidi „Pretvornici kodova“ u knjizi na str. 146-149

¹⁵ Ovo „K“ dolazi od prezimena gospodina Mauricea Karnaugh, vidi [Karnaugh map - Wikipedia](#)

Integrirani digitalni krugovi i sklopovi (knjiga):

→ Kombinacijski sklopovi

– **Koder** (knjiga str. 142-143)

- Koliko (maksimalno) ulaza na koliko izlaza
- Koliko ulaza može biti aktivno istovremeno?
- Prioritetni koder i „*don't care*“ (X) vrijednosti
- Primjer: dekadski/BCD ili dekadski/7-seg

– **Dekoder** (knjiga str. 144-146)

- Koliko izlaza na koliko ulaza; *enable* ulaz
- Simbol i struktura sklopa
- Kaskadiranje dekodera (dekodersko stablo)

– **Multipleksor** (knjiga str. 149-150)

- Svi ulazi i izlazi
- Simbol i oznake sklopa
- Kaskadiranje multipleksora (multipleksorsko stablo)

– **Demultipleksor** (knjiga str. 151-152)

- Ista struktura kao dekode... koji ulaz/izlaz je tu što (u odnosu na dekode)
- Kaskadiranje demultipleksora (demultipleksorsko stablo)

→ Matematičko-logički sklopovi

– **Zbrajalo** (knjiga str. 132-139)

- Poluzbrajalo (half-adder)
- Puno zbrajalo (full-adder) – izravno i preko poluzbrajala
- Višebitno (paralelno) zbrajanje – kaskadiranje zbrajala i preljev (*overflow*)
- Prikaz negativnih cijeli brojeva (2. komplement) i oduzimanje pomoću zbrajanja

– **Komparator** (knjiga str. 141)

- Ideja i kaskadiranje

– **ALU** (knjiga str. 139-140)

- Simbol skupa s ulazima i izlazima (prikaz višebitnih ulaza)
- Koliko bitova imaju operandi i rezultat (što je „riječ“ tj. WORD)

- Što su operandi i kako se bira operacija
- Što su statusni bitovi (tj. zastavice), primjer jedne

→ Memorijski sklopovi

– Bistabili (knjiga str. 90-99)

- Namjena/svrha
- SR – građa/ideja (što je negativna povratna veza) i simbol sa ulazima/izlazima
- JK – što popravljaju u odnosu na SR
- D, T – koliko ulaza i što rade (glavno svojstvo)
- Upravljački ulaz, koji signal tamo dovodimo najčešće, simboli i objašnjenje za padajući/rastući brid
- Asinkroni ulazi – značenje i svrha

– Registri (knjiga str. 110-120)

- Građa (koji bistabili, koliko njih itd.) i namjena (općenito)
- Različite vrste i namjene (samo idejno, struktura nije potrebna)
- Univerzalni registar – simbol i razni ulazi (S/P, L/R, R/W, ...) - objašnjenje

– Brojila (knjiga str. 120-130)

- Građa (koji bistabili, koliko njih itd.) i namjena (općenito)
- Različite vrste i namjene (samo idejno, struktura nije potrebna)

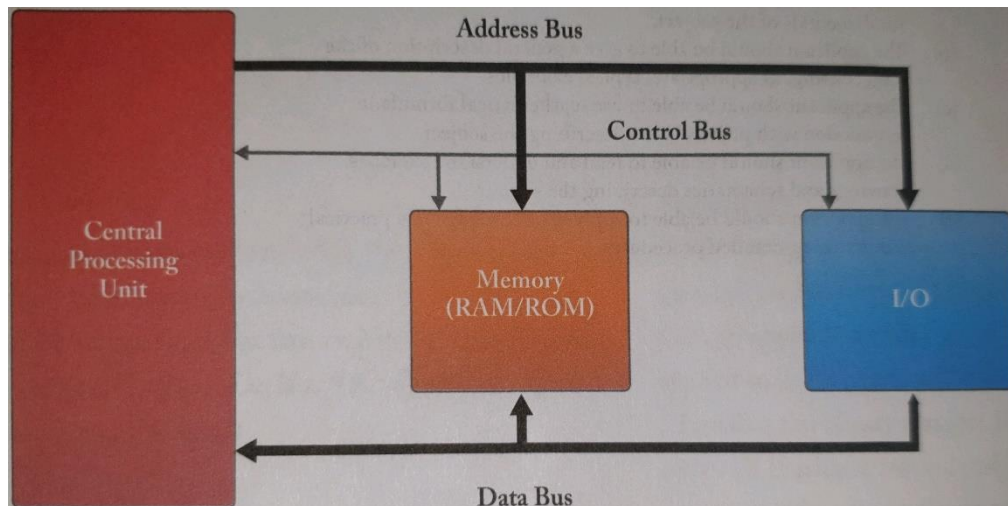
– Memorije (knjiga str. 160-178)

- Osnovna organizacija memorije (dekoder + polje + r/w međusklop)
- Karakteristični parametri (kapacitet, vrijeme pristupa, max r/w frekvencija)
- Povećanje kapaciteta memorije (načini spajanja više memorijskih čipova)
- Vrste/podjele memorije (ima raspisano tu u sljedećem odjeljku)
- Logička struktura (dekoder + koder) i vrste ispisnih memorija
 - ROM kao programirljiva komponenta (samo princip)
- Građa i vrste RAM memorija

Osnove arhitekture računala:

→ Osnovne ustrojstvene jedinice

- Osnovni dijelovi računala su: procesor (**CPU**), glavna radna memorija (**RAM**), različite **ulazno-izlazne (I/O) jedinice** te različite **sabirnice** (*engl. bus*) posebne namjene koje ih sve povezuju



– **Processor (CPU – Central Processing Unit)**

- izvršava (jednostavne) naredbe (instrukcije) jednu po jednu, što znači da pokreće sve druge komponente/sklopove u računalu, odnosno pogoni računalu – *mozak računala*
- glavni dijelovi koje sadrži svaki procesor su:
 - Takt (**CLK, CP – Clock Pulse**) – pravokutni signal po kojem se ravna rad svih drugih komponenti/sklopova; najčešće rad sklopova bude okidan na rastući ili padajući CP signal (*rising/falling edge*), dakle pri prelasku iz „0“ u „1“ ili obrnuto
 - Kontrolna jedinica (**CU - Control unit**) – u njoj se naredbe iz memorije dekodiraju i tako dobivaju signali za upravljanje svih drugih dijelova CPU-a i računala; dakle svim drugim sklopovima „govori“ što i kada (kojim redoslijedom) trebaju raditi
 - Aritmetičko logička jedinica (**ALU - Arithmetic-logic unit**) – obavlja sve aritmetičke (zbrajanje, oduzimanje, ...) i logičke (logičko i, ili, ...) operacije između binarnih podataka koji su do nje dovedeni negdje iz nekakve memorije
 - Najčešće, tj. u pravilu kod modernih procesora postoji zaseban sklop za rad s decimalnim (*floating-point*) brojevima, tzv. matematički koprocesor odnosno **FPU (Floating-point unit)**; takav sklop uvelike ubrzava i poboljšava rad s decimalnim brojevima, ali nije obavezan (pa se najčešće ne spominje)
 - **Registri** – privremeni spremnici raznih podataka koji su trenutno ili su donedavno bili ili će u neposrednoj budućnosti biti potrebni za izvršavanje instrukcija

- Postoje opći i specijalni: opći su za općenite podatke (adrese i vrijednosti operanada, rezultata operacija i slično), a specijalni imaju svoje posebne nazive i sadrže točno određene podatke (npr. instrukciju koja se trenutno izvršava, adresu slijedeće instrukcije, statusne bitove zadnje izvršene operacije itd.)
- Opći (i većina specijalnih) u pravilu su duljine jedne „riječi“ (WORD), a to je određeno arhitekturom računala, npr. 32 bita za starija i 64 bita za novija osobna računala (PC), ali može biti i bilo koji drugi broj kod specijaliziranih računala (npr. 8 ili 16 ili 20 bitova), a može biti i različito za adrese i za podatke
- Moderniji/kompleksniji procesori, ovisno o namjeni i arhitekturi, sadrže još razne druge specijalizirane dijelove među kojima se ističu:
 - **Cache** (L1/L2/L3) – relativno male količine (kilobajti do nekoliko megabajta) posebne, super-brze memorije koja se nalazi u samom procesoru i značajno ubrzava rad procesora (jer je „odlaženje“ po podatak u glavnu memoriju jako sporo, iz perspektive procesora)
 - Složena je hijerarhijski, po razinama/levelima (L1 = prva razina, L2 = druga razina itd.); što je razina niža (manji broj) to je manja količina memorije i ona je brža (skuplja) i bliža ALU
 - Ideja: iz glavne radne memorije se dohvaćaju blokovi, a podatci i instrukcije koje se češće koriste (dakle ponavljaju, a to je često u programima) spremaju se na niže razine *cachea*; procesor potrebne podatke i instrukcije traži prvo u najnižoj razini *cachea* pa onda u višima i tek ako ih u *cacheu* nema ide po njih u glavnu radnu memoriju (što je sporo i to se manifestira kao kratki zastoji i „trzavost“ u radu računala)
 - Najniža razina *cachea* (L1) u pravilu je razdvojena na dva fizički i logički zasebna i neovisna dijela – instrukcijski (*instruction cache*) i podatkovni (*data cache*)
 - Jedinica za upravljanje memorijom (**MMU** – *Memory Management Unit*) – služi za efikasnije korištenje memorije i brži rad procesora, posebno kada procesor sadrži *cache* memoriju
 - U principu dijeli memoriju na blokove (koji se učitavaju u *cache*) i obavlja funkciju prevođenja logičkih adresa (lokalne adrese podataka unutar njihovog bloka) u fizičke adrese (globalne adrese podataka u cijeloj memoriji)
 - Ukratko – zna gdje se što nalazi i kako do toga najefikasnije doći
- Svi nabrojani dijelovi zajedno čine jednu **jezgru** (execution **core**). Moderni procesori u osobnim računalima u pravilu sadrže više jezgri (2, 4, 8, ...) i postižu ubrzanje rada na način da te jezgre rade istovremeno (paralelno)
 - za to je potrebno komplicirano usklađivanje rada, da ne bi više jezgri radilo istovremeno po istim podacima, jer to rezultira krivim rezultatima i mogućim

totalnim zastojem rada (rušenjem računala); to se postiže dodatnim posebnim i kompliciranim dijelovima/jedinicama

- to je postala taktika ubrzavanja rada računala kada je daljnje povećavanje brzine takta postalo problematično iz perspektive potrošnje energije i zagrijavanja čipova (slično kao i motori s unutarnjim izgaranjem)

– Memorija

- Memorije općenito se mogu razlikovati/podijeliti po više kriterija:
 - 1) Kako se pristupa memoriji, odnosno može li se svakoj memorijskoj lokaciji pristupiti **direktno** (preko adrese) ili se mora ići **sljedno** (npr. magnetske kazete)
 - 2) Može li se u memoriju i pisati i čitati (**upisno-ispisna**, *read/write* – *rw*) ili samo iz nje čitati (**ispisna**, *read-only* – *ro*)
 - 3) Je li memorija **trajna** (*non-volatile*) ili gubi informacije bez struje (*volatile*)
 - 4) Je li memorija nužan logički dio računala i u direktnoj komunikaciji s procesorom (**unutarnja**) ili je dodatna i opcionalna (**vanjska**)
 - 5) Prema tehnologiji izrade (magnetske, optičke, poluvodičke, ...) itd.
- Postoji nekoliko osnovnih vrsta memorija u gotovo svakom računalu:

→ Radna memorija - RAM (*Random Access Memory*)

- to je unutarnja memorija, služi za pohranu podataka i instrukcija (programa) tijekom rada računala
 - zbog toga je relativno brza i ima relativno ograničen kapacitet, odnosno puno je brža (100 do 1000 puta) i puno manjeg kapaciteta (100 do 1000 puta) od velikih vanjskih memorija poput tvrdih diskova
- RAM ima direktno adresiranje i upisno-ispisna (*rw*) je (i u pravilu nije trajna, dakle podatci se gube kada se računalo ugasi)
- dvije su osnovne RAM tehnologije:
 - SRAM – napravljena od bistabila; veća brzina, ali i veća veličina i cijena po bitu; koristi se za *cache* memoriju unutar procesora
 - DRAM – jednostavnija (1 tranzistor i 1 kondenzator po bitu), manje veličine i jeftinija, ali sporija; koristi se za glavnu radnu memoriju (tipično 4, 8, 16, ... GB)

→ Trajna memorija - ROM (*Read-Only Memory*)

- to je vanjska memorija koja u pravilu služi za trajnu pohranu pogonskih programa računala (koji se tijekom rada onda učitaju u radnu memoriju)
 - tu je zapisan *firmware* jednostavnih ugradbenih računala i uređaja (više od tome u sljedećem poglavlju)

- također u takvom ROM-u je zapisan i BIOS osobnih računala (ono prvo što se pokreće kada upalite svoje računalo; također više o tome u idućem poglavlju)
- ROM također ima direktno adresiranje, ali je samo ispisna (ro) i trajna (inače ne bi imala smisla)
- sporija je od RAM-a, ali puno sitnija i brža od drugih trajnih memorija poput tvrdog diska, zato je prikladna za mala računala i uređaje
- nekoliko je različitih tehnologija:
 - PROM (*programmable ROM*) – u njega se, posebnim postupkom, može samo jednom nešto upisati i to je trajno
 - EPROM (*erasable programmable ROM*) – sadržaj memorije se, posebnim postupkom pomoću svjetlosti, može izbrisati i onda ponovno isprogramirati, ali taj proces je jako spor
 - EEPROM (*electrically erasable programmable ROM*) – sadržaj memorije može se relativno jednostavno i brzo, naponski i strujno, promijeniti (izbrisati i nešto drugo upisati)
 - **Flash ROM** – napredni EEPROM koji dozvoljava puno ciklusa brisanja i pisanja (do 100 000) i to poprilično brzo (10 do 100 puta brže od mehaničkog, magnetskog tvrdog diska)
 - Danas standardna tehnologija za mala ugradbena računala, tzv. *USB stick* memorije pa i velike vanjske trajne memorije (SSD) koje lagano zamjenjuju mehaničke, magnetske tvrde diskove

→ Ostala vanjska, trajna memorija

- Velike količine memorije koja je spora – trajno spremanje podataka koje procesor ne koristi stalno nego dohvati u radnu memoriju kad mu treba (i spremi promjene u nju kad je gotov s tim podacima, ako nije *read-only*)
- Zbog velike količine memorije, sporosti i fizičkih/tehnoloških ograničenja nemaju u potpunosti direktno adresiranje (svakog pojedinog bajta) nego druge načine logičkog upravljanja podacima – datotečne sustave (*filesystems*), dakle datoteke i direktoriji; npr. NTFS (windows), EXT4 (linux), FAT32 (opće – USB stickovi), ...
- Tipične tehnologije:
 - read/write: mehaničko-magnetski tvrdi diskovi (HDD – *Hard Disk Drive*) i poluvodički, dakle od *flash* memorije (SSD – *Solid State Drive*, USB stickovi)
 - read-only: optički (CD, DVD, Blu-ray), magnetski (kazete i diskete - davno), ...

– Ulazno/izlazne jedinice (I/O – *Input/Output Units*)

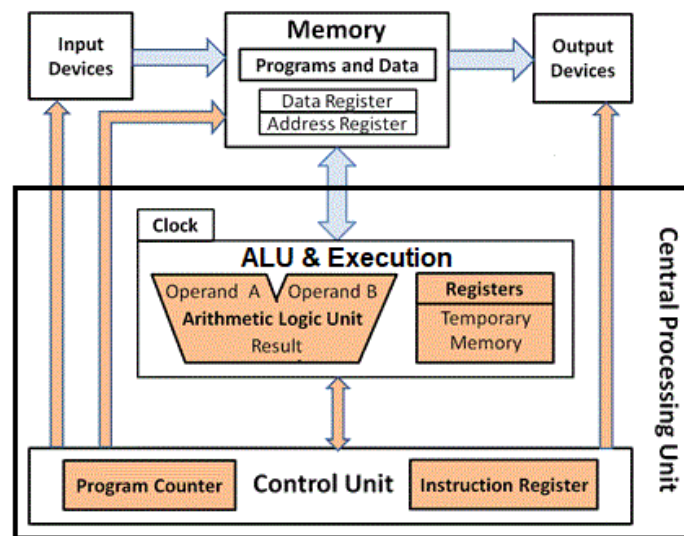
- Sve s čime, odnosno preko čega računalo, odnosno procesor komunicira s vanjskim svijetom
- Ulazne jedinice – pretvaraju neki fizički podražaj/signal/fenomen u digitalne signale (bitove) i šalju to procesoru u formatu kojeg će procesor „razumjeti“
 - Može biti izravno, dakle samo „sirovi“ bitovi iz A/D konvertera; npr. kod posebnih uređaja, malih ugradbenih računala s posebnom namjenom (razni senzori – fotodioda, mikrofona, antena, žiroskop, akcelerometar, voltmetar, ampermetar, prekidači, gumbi, tipkala, ...)
 - Može biti na „višoj“ razini, već obrađeni digitalni signali/poruke od malih uređaja koji imaju svoje mikroprocesore (ugradbena računala) u nekom točno određenom, dogovorenom obliku (protokol, *driver* program); npr. Miš, Tipkovnica, GPS, radar itd.
- Izlazne jedinice – pretvaraju digitalne signale (bitove) iz procesora u neki fizički podražaj/signal/fenomen
 - Može biti izravno, dakle direktno napon u fizikalni fenomen; npr. kod posebnih uređaja, malih ugradbenih računala s posebnom namjenom (paljenje LED dioda, pomicanje kazaljki ili drugih mehanizama itd.)
 - Može biti na „višoj“ razini, dakle bitovi od procesora su zapravo naredbe mikroprocesoru tog nekog kompleksnijeg izlaznog uređaja u nekom točno određenom, dogovorenom obliku (protokol, *driver* program); npr.
 - **grafička kartica** – izlazna jedinica kojoj procesor šalje podatke za prikazivanje na ekranu
 - u današnje vrijeme grafičke kartice su kao zasebna mala računala (imaju i svoj zasebni video RAM – VRAM – i to i gigabajtima) i onda komuniciraju s monitorima, dakle monitori nisu (izravne) izlazne jedinice vašeg računala
- Ulazno/izlazne jedinice – rade obje funkcije, npr.
 - **zvučna kartica** – ulazno/izlazna jedinica koja pretvara digitalne informacije od procesora u analogni signal za zvučnike (dakle kao i kod monitora, zvučnici nisu zapravo izravna izlazna jedinica vašeg računala), ali i analogni signal od mikrofona u digitalni signal kojeg procesor „razumije“ i može obraditi
 - **mrežne kartice** – ulazno/izlazne jedinice koje računala koriste za komunikaciju sa različitim mrežama preko različitih mrežnih protokola i medija (Ethernet, WiFi itd.)

– Sabirnice (*Buses*)

- To su skupine žice kojima digitalni signali putuju unutar računala, dakle koje povezuju različite komponente računala
 - **širina sabirnice** (*bus width*) je naziv za broj „paralelnih“ žica u sabirnici
- Tri su osnovne vrste sabirnica u računalu: kontrolna, adresna i podatkovna
 - **Kontrolna sabirnica** je naziv za žice kojima putuju signali iz i prema upravljačke jedinice (*control unit*), dakle preko kojih CU upravlja sa svim drugim sklopovljem u računalu
 - Izvedba i svi parametri ovih sabirnica su vrlo specifični za svaki procesor, odnosno familiju, arhitekturu i proizvođača procesora
 - **Adresna sabirnica** su žice koje povezuju CPU i memoriju i na koje procesor postavlja bitove koji aktiviraju određenu memorijsku lokaciju u memoriji (bili za čitanje, bilo za pisanje)
 - kod 32-bitnih računala adrese su 32-bitne (što znači adresna sabirnica od 32 paralelne žice), a to znači da postoji maksimalno $2^{32} = 4$ Gi adresa, šta pak znači da takva računala mogu imati maksimalnu radnu memoriju od 4 GB
 - zato su moderna računala relativno nedavno postala 64-bitna, da mogu podržavati (puno) veću količinu radne memorije
 - **Podatkovne sabirnice** su skupine žica kojima putuju podatci između memorije i procesora, odnosno procesora i ulazno/izlaznih jedinica
 - njihova „širina“ (dakle broj žica) ovisi o arhitekturi računala, tj. procesora, i jednaka je „riječi“ procesora (WORD)
 - npr. kod 32-bitnih računala riječ je 32-bitna, dakle 4 B, dok je kod 64-bitnih računala 64-bitna, dakle 8 B

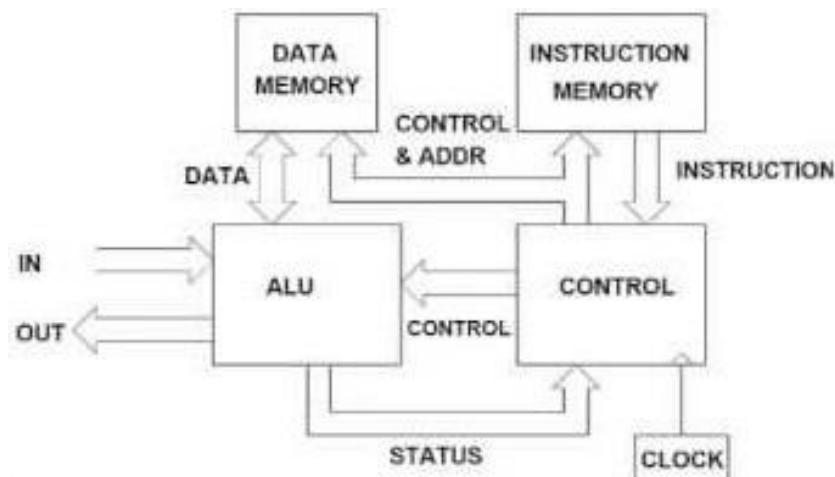
→ Arhitekture (organizacija) računala:

– Von Neumann arhitektura



- Postoji samo jedna memorija u kojoj se nalaze i instrukcije (programi) koje procesor treba izvršavati i svi podatci s kojima i nad kojima procesor radi
- Prednosti: jednostavnija i jeftinija izvedba (dovoljno dobro za većinu namjena, pogotovo jednostavnih)
- Mane: pristup memoriji je usko grlo, ne mogu se istovremeno dohvaćati i izvršavati instrukcije; moguće je „hakiranje“ programa zapisivanjem podataka u memoriju (jer ju dijele)

– Harvard arhitektura



- Postoje dvije odvojene memorije – instrukcijska i podatkovna
- Prednosti: mogućnost istovremenog čitanja i izvršavanja instrukcija, veća brzina, zaštita od neželjenog utjecaja programa jednog na drugi

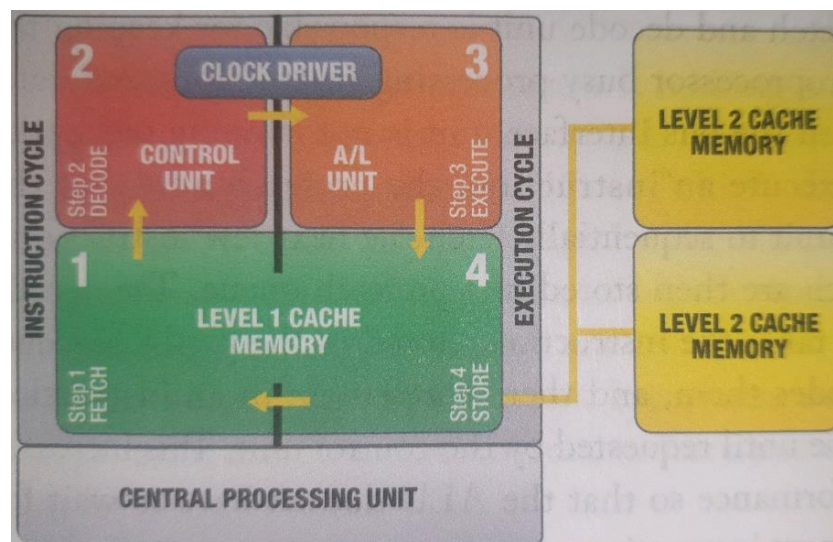
- Mane: kompliciranija i skuplja izvedba, u pravilu se radi samo kada donosi ubrzanja koja se ekonomski isplate

– **Hibridna arhitektura** (na primjeru modernih procesora za osobna računala)

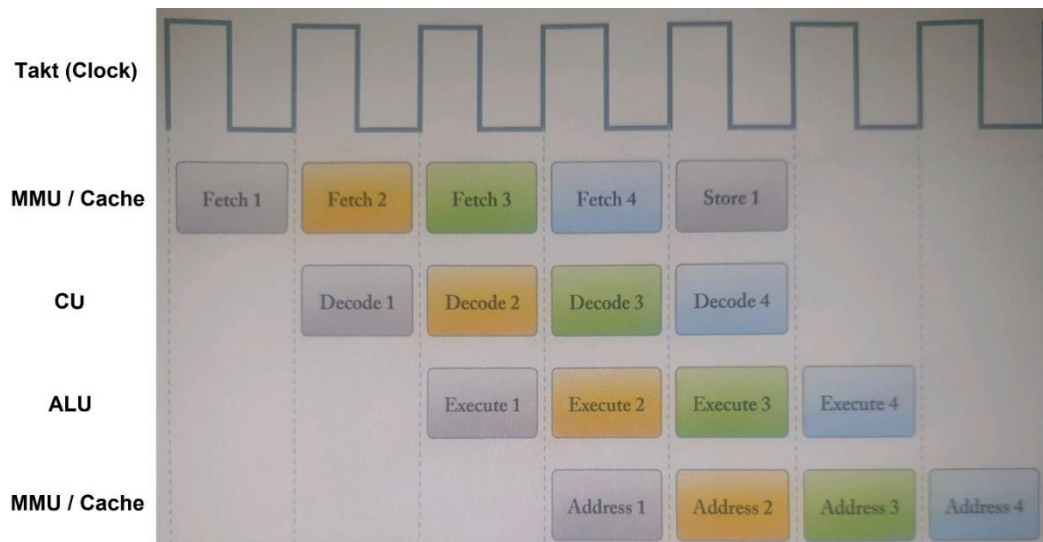
- Na razini računala postoji samo jedna radna memorija u kojoj su spremljene i instrukcije i podatci → *Von Neumann arhitektura*
 - Zapravo dvije su memorije, jedna trajna i spora (HDD ili SSD) i jedna ne trajna, ali puno brža (RAM) u kojoj se nalazi ono po čemu se trenutno radi i iz koje se promjene spremaju nazad na trajnu memoriju jednom kada je procesor gotov s tim podacima na neko duže vrijeme (npr. kad program završi)
 - RAM-a treba biti dovoljno da podnese sve što se trenutno radi – ako ne može onda računalo postane JAKO sporo tj. *unresponsive* jer počne koristiti trajnu memoriju kao RAM, a ona je 100-1000x sporija
- Na razini procesora postoji odvojena L1 instrukcijska i podatkovna *cache* memorija → *Harvardska arhitektura*
 - L1 je količinski najmanja, ali i najbrža memorija koja je najbliža procesoru – značajno ubrzava rad računala

→ Arhitektura i osnove rada mikroprocesora:

- Svaki procesor ima neke osnovne dijelove (kako je prikazano na slici ispod) i svaki od njih obavlja drugačiju funkciju
- Izvršavanje instrukcija od strane procesora može se podijeliti na korake koje izvršavaju različiti dijelovi procesora, npr:
 - 1) Dohvati (sljedeću instrukciju iz memorije) – to obavljaju *Cache* i MMU
 - 2) Dekodiraj (tu dohvaćenu instrukciju) – to radi kontrolna jedinica (CU)
 - 3) Izvrši (tu naredbu) – to radi ALU (i eventualni dodatci posebne namjene poput FPU)
 - 4) Spremi (rezultat ALU) – to obavljaju opet *Cache* i MMU, ali to se može paralelno izvršavati s korakom 1 baš zato jer su instrukcijska i podatkovna memorija odvojene na L1 razini (harvardska arhitektura)



- Ako sve naredbe/instrukcije imaju tu istu strukturu koraka onda se njihovo izvođenje može značajno ubrzati korištenjem tehnike **cjevovoda (pipeline)**
 - Ideja: dok jedan dio procesora „obrađuje“ svoj dio jedne naredbe drugi dijelovi procesora mogu „obrađivati“ svoje dijelove drugih (budućih/sljedećih) naredbi
 - Na taj način može se postići veliko, pa i maksimalno ubrzanje, u smislu da iako procesoru treba nekoliko taktova za obraditi i izvršiti jednu cijelu naredbu, u prosjeku se izvršava jedna cijela naredba po taktu (vidi sliku ispod)
- Na osnovu toga procesori se dijele na dvije osnovne vrste arhitekture procesora: CISC i RISC
 - **CISC (Complex Instruction Set Computer)** nudi korisnicima (programerima) veću količinu kompliciranijih operacija/naredbi/instrukcija koje izvršavaju posebni „moduli“, dakle dijelovi procesora specijalizirani za brzo i efikasno izvršavanje



specifično tih operacija (kao npr. FPU za aritmetičke operacije nad decimalnim brojevima, ali uglavnom još kompliciranije)

→ Pozitivna strana je brzina provedbe tih posebnih naredbi kao i jednostavnost programiranja

→ Negativna strana je kompliciranost (pa onda i cijena) takvog procesora kao i nemogućnost većeg/maksimalnog ubrzanja koristeći tehniku cjevovoda

→ Primjer: *Intel procesori* (osobna računala itd.)

- **RISC (Reduced Instruction Set Computer)** nudi korisnicima vrlo malo količinu (desetak) najosnovnijih naredbi koje sve imaju istu strukturu koraka

→ Pozitivna strana je mogućnost velikog/maksimalnog ubrzanja procesora korištenjem tehnike cjevovoda

→ Negativna strana je kompliciranost u programiranju i optimizacija za specifične namjene

→ Primjer: *ARM procesori* (mali uređaji poput pametnih telefona, Apple računala, ...)

Programska podrška (softver):

→ **Definicija:** skup programa (naredbi) i podataka koje govore računalu što će i kako raditi, tj. koji služe korisniku da upravlja s računalom (hardverom)

→ Programski jezici (TODO):

- Strojni jezik i assembler (niži programski jezici) ...

- Viši programski jezici...

- ...

- Prevođenje programskih jezika (kompajliranje)

- ...

- Interpretirani jezici...

- Java (?)

→ Dvije su osnovne vrste programske podrške: sistemska i aplikacijska

- **Sistemska programska podrška** je ona koja omogućava rad sustava kao takvog, dakle ona koja sustavu daje njegove osnovne funkcije i koja omogućava iskorištavanje hardverskih mogućnosti (resursa) od strane korisnika i drugih programa; vrste su:

- **Firmware** – jednostavni pokretački programi malih ugradbenih uređaja

- nešto kao jednostavni operacijski sustavi jednostavnih ugradbenih sustava (mikrokontrolera itd.)

- upisan trajno u ROM takvih uređaja (danas najčešće *flash* memorija)

- u pravilu se može promijeniti/ažurirati, ali se to radi rijetko – *flashanje*

- primjer: BIOS (*Basic Input-Output System*)

- to je ono što se pokreće pri paljenju računala i što onda pokreće „veliki“ operacijski sustav sa tvrdog diska (učitava ga u radnu memoriju)

- dolazi unaprijed i trajno zapisan na zasebnoj ROM memoriji pri kupovini računala (negdje na matičnoj ploči)

- **Operacijski sustavi** – nakupina programa i alata koji služe kao sistemska podrška kompliciranim računalima opće namjene

- Sastoje se od:

- **jezgre (kernel)** - pruža sve osnovne funkcije poput upravljanja memorijom, pokretanja i izvođenja drugih programa (procesa), više njih istovremeno, sigurnosti i ograničavanja pristupa itd.

- **pomoćnih programa** – skupina osnovnih programa za korištenje računala poput pregleda i pretraživanja memorije (tekstualnog i grafičkog), pokretanja i gašenja drugih programa i procesa, stvaranja i brisanja datoteka i direktorija,

otvaranja i prikazivanja datoteka različitih vrsta, upravljanja različitim vanjskim ulazno/izlaznim jedinicama (*driveri*) itd.

- **Driveri** (pogonski programi) su programi koji služe za komunikaciju i upravljanje ulazno/izlaznim jedinicama od strane računala, dakle to su programi koje kad procesor izvršava onda prima i šalje podatke ulazno/izlaznim jedinicama tako da on „razumije“ njih i one njega

→ Primjeri: **Windows** (razne verzije, najnovija je 10), **Linux** (razne distribucije npr. Ubuntu), **macOS** (za Apple računala), **Android** (za mobilne uređaje – baziran na Linux jezgri i Java jeziku) itd.

– **Aplikacijska programska podrška** su svi ostali programi koji se nalaze na računalu, a koje korisnik može pokretati (kroz operacijski sustav) da bi koristio računalo na neke naprednije i specifične načine, bez da su sami nužni za rad i korištenje računala

- Primjeri: Microsoft Office alati, preglednici (*browseri*, npr. Google Chrome, Microsoft Edge, ...), multimedijски programi, profesionalni alati/programi (npr. Photoshop), igrice itd.

→ Programska podrška za zrakoplove:

- Moderni zrakoplovi imaju puno računala, praktički su jedno veliko računalo, i sva ta računala izvršavaju neki kôd, tisuće i milijune linija kôda kojeg su napisali neki ljudi sjedeći za nekim stolom (a ljudima postane dosadno i izgube koncentraciju)
- Sav taj kôd napisan je posebno za zrakoplove jer sva zrakoplovna računala su posebna i imaju posebnu programsku podršku
- Računarstvo je poznato po tome da programi sadrže dosta grešaka (*bugova*) i da je teško izbjeći njihov nastanak, a da je posebno teško pronaći ih ili uopće utvrditi da postoje prije nego uzrokuju neki problem
- Budući da bi takve pogreške kod zrakoplova mogle dovesti do katastrofalnih posljedica (Googleaj padove i prizemljenja Boeinga 737 MAX u 2019. godini) američka i europska agencija izradile su standard za postupak izrade i provjere programske podrške za zrakoplovne računalne sustave kako bi ispunili zahtjev plovidbenosti (*airworthiness*)
- Među ostalim taj standard (imenom DO-178B/ED-12B) definira 5 razina kritičnosti neotkrivene pogreške u kodu u smislu posljedica koje bi to uzrokovalo:

A) **Katastrofična** (*Catastrophic*) – onemogućuje (siguran) let i/ili slijetanje

→ Primjeri sustava na kojima bi greška bila katastrofična: AHRS, GPS/ILS/MLS/FLS, SATNAV, VOR, ADF

B) **Opasna/teška** (*Hazardous/Severe*) – velike negativne posljedice na sigurnost leta

→ Primjeri sustava na kojima bi greška bila opasna/teška: TCAS, ADBS transponder

C) **Velika** (*Major*) – uvelike smanjuje sigurnost i uzrokuje neudobnost i moguće ozljede

→ Primjeri sustava na kojima bi greška bila velika: DME, VHF glasovna komunikacija

D) **Mala** (*Minor*) – stvara samo (manje) neugodnosti

→ Primjeri sustava na kojima bi greška bila mala: AHRS, CMC/CFDIU, radar za vrijeme

E) **Bez efekta** (*No Effect*)

→ Primjeri sustava na kojima bi greška bila bez efekta: sustav za zabavu putnika

- Jednom kada bi *software* dobio odobrenje za korištenje, odnosno potvrdu plovidbenosti, nikakvo ažuriranje i promjene na njemu nisu moguće bez posebnog odobrenja ako bi moguće greške u promjenama spadale u razine kritičnosti A, B ili C.

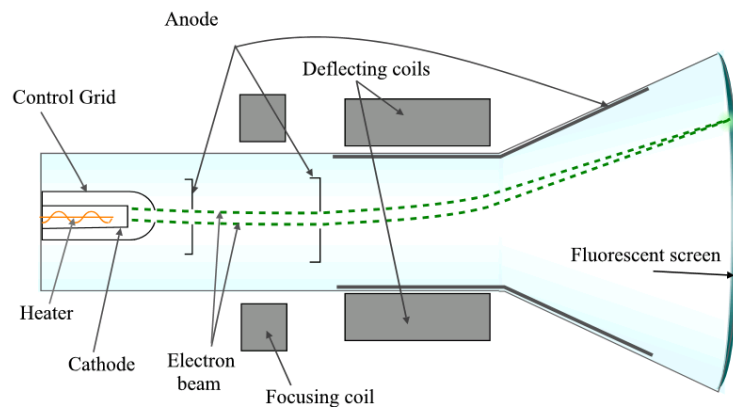
Elektronički zasloni:

→ Osnovne tehnologije

– Zaslони s katodnom cijevi (Catode-Ray Tube - CRT)

- Stara tehnologija, koristila se za prve televizore te računalne i druge monitore (npr. osciloskopi i razni instrumenti)
- U zrakoplovstvu je bila osnova tzv. „staklenih kokpita“ od ranih 1970-ih, dakle koristili su se u prvim zrakoplovima s elektronikom i još uvijek se mogu pronaći u starijim modelima zrakoplova
- Princip rada:

→ Katodna cijev je (relativno) dugačka, staklena, vakumirana cijev kroz koju putuje mlaz elektrona (elektronska zraka)



→ Zagrijavanjem metalne žice (katoda) stvaraju se slobodni elektroni koji se onda velikom razlikom potencijala (anoda) ubrzavaju prema drugom kraju cijevi (zaslon)

– Razlika potencijala (napon) može biti i do 20 kV

→ Na tom putu (od katode do zaslona) elektronski snop se oblikuje i usmjerava i to na jedan od dva moguća načina:

1) parovima pozitivnih i negativnih pločastih elektroda unutar vakumske cijevi (elektrostatski)

- ovaj način daje precizniju sliku i ekran može biti ravan, ali manji – koristi se kod instrumenata s malim ekranima (npr. osciloskop)

2) zavojnicama omotanima oko vakumske cijevi (magnetski)

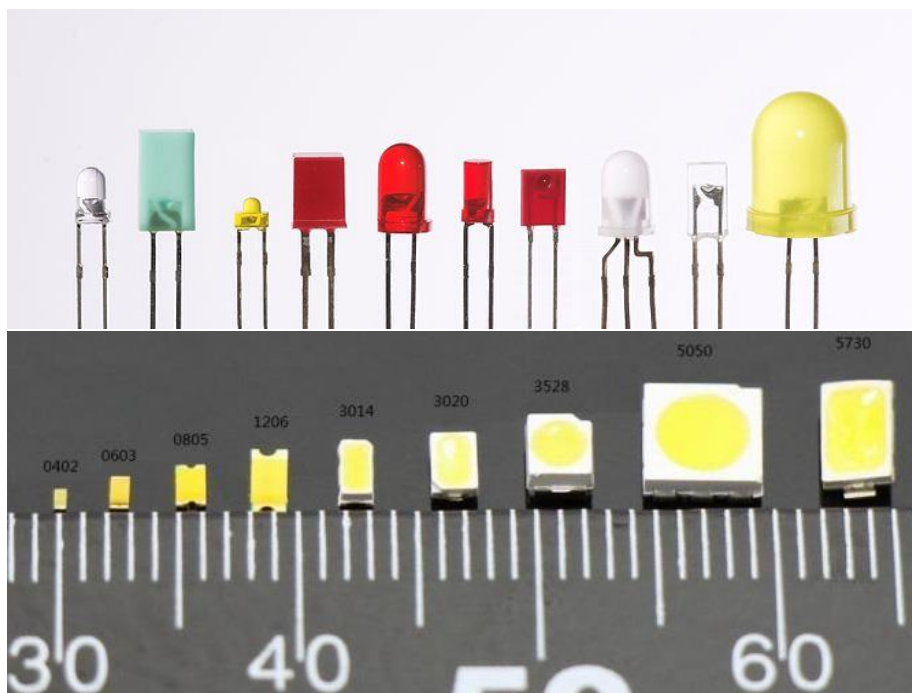
- ovaj način omogućuje veće ekrane s ne prevelikom „dubinom“ – tipična uporaba kod televizora i velikih računalnih monitora

→ S unutarnje strane zaslona nalazi se premaz (od fosfora) koji zasvijetli tamo gdje elektroni iz snopa u njega udaraju

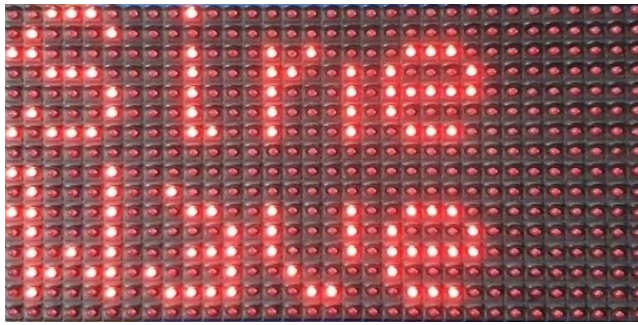
- Intenzitet svjetlosti ovisi o brzini elektrona, dakle o naponu koji ih ubrzava
- Snop se fokusira i udara u pojedine točke – **pikseli**; u svakom trenutku svijetli samo jedan piksel (elektronski snop pogađa jednu točku)
- Postoje 2 osnovna načina rada:
 - 1) **Raster** – elektronska zraka prelazi redak po redak s lijeva na desno (kao skener) i tako popunjava cijeli ekran (sve piksele)
 - to je tipičan način rada televizora i računalnih monitora, idealan za popunjavanje pozadine, odnosno da cijeli ekran „svijetli“
 - da bi se dobio dojam da ekran svijetli cijeli odjednom (bez treperenja) potrebno je da elektronski snop prođe sve piksele barem 30 puta u sekundi – to se zove *refresh rate*
 - tipičan *refresh rate* za CRT ekrane je 60 Hz (60 puta u sekundi), a standardna rezolucija u ono vrijeme je bila 1024x768, što znači 768 redaka po 1024 piksela, sve skupa 786 432 piksela; ako uzmemo da je za obradu jednog piksela potreban jedan period takta procesora, tj. grafičke kartice, onda to znači da su računala morala raditi na minimalnoj frekvenciji od $786\,432 \cdot 60 \approx 50$ MHz, odnosno realističnije nekoliko stotina MHz, da bi mogla podržavati CRT ekrane takve rezolucije
 - 2) **Vektorska grafika** ili potez (*stroke*) – elektronska zraka crta krivulje i oblike slobodnim kretanjem po cijelom ekranu
 - koristi se za crtanje simbola i drugih sličnih stvari preko neke pozadine – vrlo često u zrakoplovnim instrumentima
 - u praksi zasebni elektronski snop za takvo crtanje, znači jedan (ili 3) za raster način + 1 zaseban snop (u istoj katodnoj cijevi) za vektorsku grafiku
- Ekran u boji postiže se tako da se svaki piksel sastoji od 3 točkice od različitih premaza (fosfor s različitim aditivima) koja svaka daje drugu boju (crvenu, zelenu i plavu – RGB) i u koje istovremeno „udaraju“ tri odvojena elektronska snopa
- Vrhunac CRT tehnologije je Sony Trinitron tehnologija koja je omogućila vrlo preciznu sliku u boji na velikim i ravnim ekranima, a sa manjom „dubinom“ ekrana, dakle kraćom katodnom cijevi
 - To je postignuto, među ostalim, i vrlo pametnom izvedbom „maske“ kroz koju prolazi elektronski snop prije udara u premaz na ekranu
- Generalne mane su veličina, težina, velika snaga tj. potrošnja električne energije te osjetljivost na elektromagnetske smetnje – sve od čega je vrlo problematično za zrakoplovstvo

– LED (Light-Emitting Diode) zasloni

- LED diode su poluvodičke komponente koje emitiraju svjetlost, a pritom troše vrlo malo energije (puno manje od bilo kojih drugih izvora svjetlosti) i imaju vrlo dug vijek trajanja
- Svaka LED dioda emitira svjetlost na točno određenoj valnoj duljini, dakle svjetlost točno određene boje
 - Crvene su bile prve, a i danas su najčešće
 - Plave su dugo bile problem, napravljene su na zadovoljavajućoj razini tek 1990-ih i za to je dodijeljena, između ostaloga, i Nobelova nagrada iz fizike 2014.; i dalje je njihova proizvodnja najkompliciranija i najskuplje su
 - Tipično dostupne su još i zelene, žute, bijele itd.
- LED diode su prekrivene nekom vrstom zaštite (epoksi smolom) i proizvode se u kućistima različitih veličina i oblika – od reda veličine centimetra sa slobodnim (dugačkim) kontaktima do SMD (za na tiskane pločice) manjima od milimetra



- Diode velike veličine koriste se kao pojedinačni indikatori (jedna lampica) ili matrice niske rezolucije (npr. 30x50 crvenih piksela), također i kod sedam-segmentnih prikazivača (displeja) – tako su se koristile i u zrakoplovima od početaka upotrebe elektronike pa sve do danas

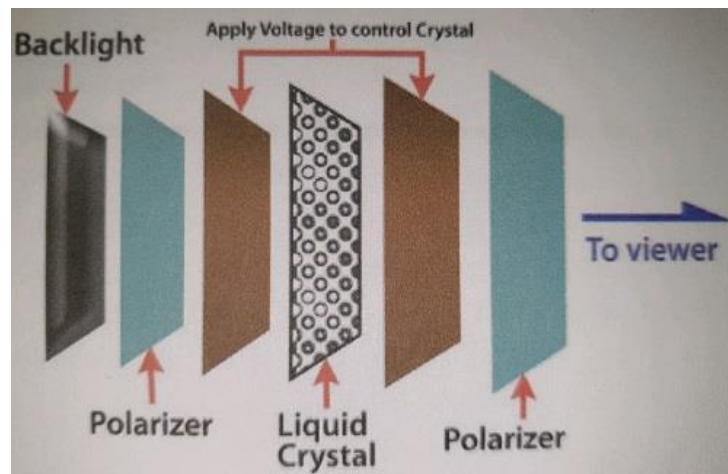


- Zbog minijaturizacije danas postoje i veliki LED displeji u boji (3 LED diode po pikselu), ali u pravilu ne u zrakoplovima nego za velike ekrane na otvorenom (npr. na stadionima)
 - Troše vrlo malo energije, imaju odlična optička svojstva, a uz to su vrlo dugotrajni i izdržljivi na vanjske uvjete – ukratko, **ako se može napraviti od LEDica onda je i najbolje** (ali ne nužno i jeftino)
- Nove tehnologije microLED i OLED omogućuju izradu potrošačkih ekrana (televizije, mobiteli itd.) visoke rezolucije, brzine i energetske efikasnosti
 - microLED je još u povojima, trenutno je proizvodnja preskupa za potrošačku upotrebu i ima ih vrlo malo, ali vjerojatno je tehnologija budućnosti jer ima fantastična svojstva
 - OLED je specifična tehnologija, te „diode“ su rađene od organskog materijala i malo drugačija je fizika, ali u načelu isto funkcioniraju i imaju vrlo slična svojstva – dakle proizvode svoje svjetlo vrlo energetski efikasno i imaju veliki kontrast i brzinu
 - „**najbolja**“ **masovna komercijalna tehnologija ekrana trenutno**
 - Proizvode se prozirni i fleksibilni ekrani, pogotovo za male uređaje poput pametnih telefona i drugih koji trebaju ta svojstva
 - AMOLED tehnologija koristi isti princip aktivne TFT matrice kao i LCD ekrani za aktivaciju pojedinih piksela što omogućuje veće rezolucije, brzine i druga bolja grafička svojstva (na račun kompliciranosti i cijene)
 - Često ide u kombinaciji s ekranima osjetljivima na dodir (*touchscreen*) i sigurno se pojavljuju i pojavljivat će se sve više u zrakoplovima, i u kokpitima i za zabavu putnika

– LCD (Liquid Crystal Display) zasloni

- Tehnologija koja je zamijenila CRT ekrane – tanki i laganiji ekrani, troše puno manje energije i imaju veće rezolucije i bolja optička svojstva
- Standardna tehnologija za televizijske i računalne ekrane svih vrsta i oblika, do pojave OLED tehnologije koja je bolja (ali trenutno još uvijek skuplja)
 - Također se koriste i za 7-segmentne i druge jednostavnije prikazivače u raznim instrumentima i na upravljačkim pločama (npr. u automobilima, pa tako i zrakoplovima)

- Za razliku od CRT i LED tehnologija koje stvaraju svoj svjetlo (emisivne su), LCD tehnologija zahtijeva vanjski, posebni izvor svjetlosti (*backlight*)
 - Kao izvor te (bije)le) svjetlosti često se koriste LED lampe zbog svoje energetske efikasnosti i dugotrajnosti/nekvarljivosti
 - Zbog te osobine, stalnog pozadinskog bijelog svjetla, LCD ekrani umaraju i štete očima pa se o tome treba voditi računa
- Princip rada:
 - Ekran se sastoji od nekoliko slojeva:



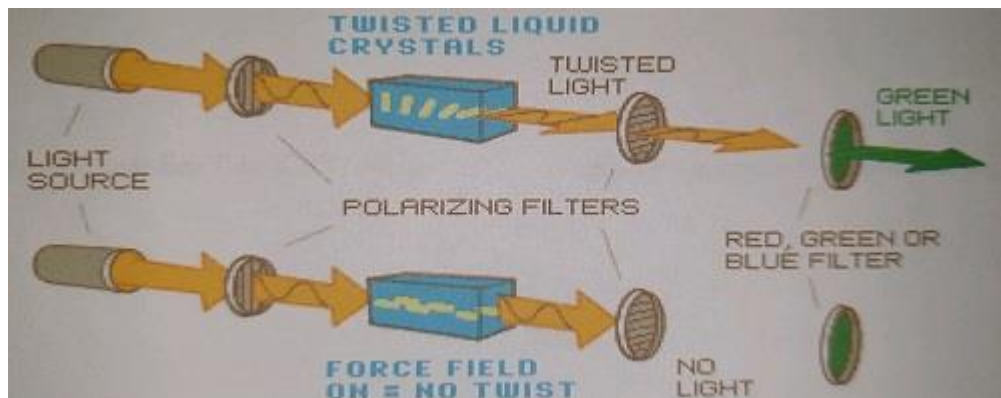
- 1) **Pozadinsko svjetlo** – postoji nekoliko različitih načina ostvarivanja te pozadinske svjetlosti, najčešće LED ili fluorescentne lampe na stražnjoj strani ili po rubovima ekrana
- 2) **Prvi polarizator** – svjetlo iz izvora je nepolarizirano, prvi polarizator polarizira to svjetlo u određenom smjeru na način da propušta samo svjetlost koja već „titra“ u tom smjeru, a blokira svu ostalu¹⁶
- 3) **Sloj „tekućeg kristala“**, tj. nematika
 - nematiki su posebna vrsta (gustog) tekućeg materijala od štapićastih molekula koji se prirodno poslože tako da zakreću polarizaciju svjetlosti koja kroz njih prolazi, a kada se na njih nametne napon onda se poslože tako da više ne zakreću polarizaciju svjetlosti koja kroz njih prolazi
 - ovisno o debljini sloja nematika, polarizacija svjetlosti se zakreće više ili manje, a za točno određenu debljinu sloja zakreće se za točno 90°
 - to svojstvo nematika osnova je ove tehnologije

¹⁶ Svjetlost su transverzalni elektromagnetski valovi, što znači da „titraju“ okomito na smjer kretanja; polarizirana svjetlost je ona sastavljena od elektromagnetskih valova koji svi „titraju“ u istom smjeru okomitom na smjer kretanja; primjer – to rade i (kvalitetnije) sunčane naočale, zato se sa nekima od njih, pod nekim kutom, ne vidi slika iz LCD ekrana

4) **Drugi polarizator** – on je postavljen tako da propušta samo svjetlost koja je polarizirana okomito od one koju propušta prvi polarizator

5) **Staklo/zaštitni sloj** (opcionalno)

→ Dakle, ono kako tehnologija radi je da za svaki piksel na kojem je nametnut napon („upaljen piksel“) sloj nematika ne zakreće svjetlost i onda ona ne prolazi do gledatelja jer 2 okomito postavljena polarizatora blokiraju svu svjetlost, a svaki piksel na kojem nije nametnut napon („ugašen piksel“) sloj nematika zakreće svjetlost za 90° i ona onda prolazi kroz drugi polarizator do gledatelja; odnosno ukratko – „upaljeni“ piksel = crni piksel, „ugašeni“ piksel = bijeli piksel



- Dvije su osnovne tehnologije LCD ekrana:

- Pasivna matrica – svaki pojedini piksel se aktivira na sjecištu jednog retka i jednog stupca, u smislu da se dovede napon na cijeli redak i cijeli stupac

- Slično kao i kod CRT ekrana, slika se na ekranu prikazuje tako da se minimalno 30 puta u sekundi „pređe“ preko svih piksela, redak po redak (kao skener)

- Ovaj način rada ne može postići visoke rezolucije (ali je jeftiniji)

- Aktivna matrica (**AMLCD**) – uz sloj nematika postoji tanka prozirna ploča sa TFT (*Thin-Film Transistor*) tranzistorima za svaki piksel i svaki pojedini piksel može se izravno aktivirati pomoću svog zasebnog tranzistora

- Ovo omogućuje veće brzine osvježavanja piksela pa su kontrast i ostala optička svojstva puno bolja (to je moderni standard)

Elektrostatički osjetljivi uređaji:

→ Statički elektricitet

- To je električno polje i električna sila koja je uzrokovana time da neki predmet nije električki neutralan nego ima više pozitivnog ili negativnog naboja (inače je sva materija, svaki atom i cijeli naš svijet - koji je sastavljen od atoma – električki neutralan, dakle ima jednako pozitivnog koliko i negativnog naboja)
- Neko tijelo (predmet, osoba, ...) postaje električki nabijeno preko trenja – doslovno fizičkom silom struganja materijala o materijal elektroni bivaju otrgnuti od svojih atoma i ono gdje završe postane negativno nabijeno (ima više elektrona), a ono odakle odu postaje pozitivno nabijeno (ima viška protona)
- Različiti materijali se različito ponašaju u tom smislu – sa nekih je lakše ili teže „otrgnuti“ elektrone, nekima je lakše ili teže primiti i privući elektrone s drugih materijala. To se zove **triboelektričnost** i postoji skala koliko je koji materijal triboelektričan, na primjer:
 - Neki pozitivno triboelektrični materijali, od veće prema manjoj triboelektričnosti:
 - **suha ljudska koža** (najlakše predaje elektrone), **koža, krzna, staklo, kosa, najlon, vuna, olovo, svila, aluminij, papir** (predaje elektrone, ali slabo)
 - **pamuk i čelik** su primjeri triboelektrično neutralnih materijala (niti predaju niti uzimaju elektrone trenjem)
 - Neki negativno triboelektrično materijali, od manje prema većoj triboelektričnosti:
 - **drvo** (privlači elektrone, ali slabo), **jantar, guma, bakar i plemeniti metali, većina plastika** (PVC itd.), **stiropor, silikon, teflon** (najviše privlači elektrone)
- Inače bi se taj viška naboj koji se nakuplja trenjem normalno praznio kroz vodljive materijale i odlazio u zemlju (zato postoji uzemljenje), ali ako je neki materijal dobar izolator onda se naboj na njemu gomila i teško s njega odlazi. Isto vrijedi i za nas ako smo obučeni u dobre izolator (npr. gumeni đonovi).
- Viška naboj se također prazni i kroz zrak, ako je dovoljno vlažan, dakle ima dovoljno vode pa je dovoljno vodljiv, ali ako je zrak suh onda ne može odvoditi viška naboj i smanjivati statički elektricitet...
- Ako se puno viška naboja nakupi, dakle nakupi se veliki statički elektricitet, i onda se dođe u kontakt s nečim vodljivim događa se naglo pražnjenje, dakle razvija se **velika struja** (puno naboja proteče u malo vremena), a to je ono što nas **peče**, uzrokuje **trzaj i oštećuje materijale**
 - Štoviše, ako se nakupilo dovoljno naboja elektricitet će može biti dovoljno velik da do kontakta i ne mora doći nego dođe do proboja zraka, dakle „skoči“ iskra!
 - A čak i bez proboja i velike struje, samo veliko elektrostatičko polje može stvarati (nevidljivu) štetu i interferenciju u osjetljivoj elektronici i sličnim uređajima...

– Primjeri nastajanja statičkog elektriciteta:

- Hodanje po vunenom ili najlonskom tepihu u obući s gumeni đonom stvara tipično 1.5 kV elektrostatskog napona pri 80% relativne vlažnosti zraka, ali čak 35 kV pri 20% relativne vlažnosti zraka, što je dovoljno da uzrokuje proboj zraka i baci „iskru“ između nas i nečega što dotaknemo
- Hodanje po podu od polivinila, s druge strane, stvara tipično samo oko 300 V elektrostatskog napon pri 80% vlažnosti, i oko 10 kV pri 20% vlažnosti zraka, što je puno manje i u pravilu nedovoljno za to da „skoči“ iskra i opeče nas

→ Elektrostatski izboji (ESD – Electrostatic discharge)

– Elektronički elementi, uređaji i komponente su JAKO osjetljive na elektrostatske izboje

– Američki vojni standard za ESD (MIL-STD-1686C) definira dvije klase elektrostatski osjetljivih uređaja:

- Klasa 1 (Class I) – od 100 do 1000 V
- Klasa 2 (Class II) – od 1000 do 4000 V

sve manje osjetljivo od toga se ne smatra elektrostatski preosjetljivim uređajima, u smislu da razumne, obične mjere zaštite (dotaknut nešto vodljivo da se „isprazniš“ i ne trljat se o nešto jako triboelektrično nego samo normalno sjedit i radit u normalnom okruženju) neće dovesti do oštećenja (najvjerojatnije)

– Elektronika bazirana na bipolarnim tranzistorima (TTL) je (kao što smo bili rekli ranije) puno robusnija, jer su bipolarni tranzistori puno otporniji i manje osjetljivi, pa TTL uređaji uglavnom spadaju u klasu 2; nažalost to je gotovo nimalo moderne elektronike

– Velika većina moderne elektronike (gotovo sva) bazirana je na MOSFET tranzistorima i CMOS tehnologiji i ona je gotovo isključivo sva klase 1!

– Problem ESD-a je s vremenom, dakle napretkom tehnologije i minijaturizacijom, sve gori i gori jer je elektronika sve osjetljivija

- To posebno vrijedi za zrakoplove jer elektronika može elektrostatskim poljima i utjecajima biti oštećena bez da je potpuno izgorila, a to može biti teško za detektirati i može uzrokovati vrlo neželjena i neočekivana ponašanja u najgorim trenucima!

→ Zaštita od ESD-a

– Rad sa ESD-osjetljivim komponentama za zrakoplov samo u kontroliranim uvjetima i uz držanje određenih pravila i procedura:

- Korištenje posebnih znakova za označavanje elektrostatske osjetljivosti



- Micanje izolirajućih triboelektričnih materijala iz okoline koliko je god moguće
- Nošenje vodljive odjeće (npr. anti-static kute) i obuće (kožni đonovi) i korištenje drugih načina odvodnje naboja sa sebe i predmeta do uzemljenja (npr. vodljivi pod)
- Pod bi trebao biti anti-statičan (polivinil, razni premazi itd.)
- Radna mjesta za rad s elektronikom (npr. stol) bi trebala imati poluvodljivu ($\approx 100 \Omega/\text{cm}^2$) radnu površinu uzemljenu preko $1 \text{ M}\Omega$ otpornika
 - Zašto? Tako da može odvoditi viška naboj sa komponenti, ali da struja izboja/odvodnje ne bude prevelika pa da ne ošteti uređaj (jer da je otpor jako mali, a ima puno nakupljenog naboja potekla bi velika struja)
- Osoba koja dira elektroniku treba nositi *anti-static* traku za zapešće koja je također uzemljena preko $1 \text{ M}\Omega$ otpornika, iz istih razloga kao i radna površina



Slika 2 Anti-static traka za zapešće

- Regulacija zraka/atmosfere u prostoru u kojemu se radi sa elektronikom
 - Održavanje vlage visokom je dosta teško, a i to su vrlo neugodni uvjeti za rad
 - Umjesto održavanja visoke vlažnosti zraka koriste se **ionizatori zraka**
 - Ionizatori zraka su uređaji koji izbacuju pozitivne i negativne ione koji onda budu privučeni suprotno nabijenim predmetima i odnose viška naboj sa njih
 - Postoje kao **pištolji** za de-elektriziranje (fokusrano, ciljano)...
 - ... ili kao uređaji instalirani iznad radnog mjesta koji stalno „pušu“ ione, poput klima uređaja



– Prijenos i skladištenje ESD-osjetljivih komponenti:

- u posebnim, vodljivim, metaliziranim vrećicama i kutijama
- na uzemljenim vodljivim policama



Elektromagnetsko okruženje i interferencija:

→ Nastajanje elektromagnetskih smetnji (EMI – electromagnetic interference)

- Sama prisutnost naboja stvara električno polje, a svako kretanje naboja (dakle svaka struja) stvara magnetsko polje svuda u vodiču i prostoru oko vodiča. Zauzvrat, svaka pojava i promjena električnog i/ili magnetskog polja uzrokuje gibanje naboja koji se u njima nalaze.
- Električna i magnetska polja i promjene (poremećaji) u njima mogu se „odvojiti“ od svojih izvora i putovati daleko kroz prostor ili vodič u obliku elektromagnetskih valova. To uzrokuje kretanje i pomicanje naboja na udaljenim mjestima.
- Kretanje i pomicanje naboja u nekom sustavu zbog rada nekog drugog sustava, odnosno EM utjecaj nekog sustava na drugi sustav koji je neželjen očituje se kao šum odnosno smetnje i to se zove elektromagnetska interferencija (**EMI**).
- Dvije su osnovne vrste izvora EMI-ja:
 - 1) Obični, normalni, „mirni“ signali (struje i naponi) u električnim i elektroničkim sustavima koji jednostavno mogu „curiti“ van tog sustava i utjecati na druge, najčešće obližnje sustave, zbog blizine njihovih vodova i slično;
 - 2) Signali s naglim promjenama (npr. pravokutni impulsi), izboji (npr. iskre i munje) i ostali slični ne-sinusoidni oblici. Takve pojave/signali sastoje se od mnogih (visokih) frekvencija i mogu stvarati šum u uređajima koji rade ili su osjetljivi na te frekvencije (antene, računala itd.). Zato je iskrenje u zrakoplovu jako nepoželjno, a mogući uzroci su: statički elektricitet, releji, četkice u DC generatorima i motorima itd.
- Većina EM zračenja je vrlo slaba (mala snaga) pa stvara mali šum od kojeg se da relativno lagano zaštititi i koji ne uzrokuje trajnu štetu, ni na opremi ni ljudima
 - Izvori su predvidljivi i unutar zrakoplova, npr. osobni uređaji (mobiteli, laptopi, ...) putnika i posade te sva ostala elektronika na zrakoplovu
- Neka EM zračenja su velike snage i mogu uzrokovati veliku i trajnu štetu na opremi, pa i na ljudima; to su tzv. **HIRF** (high-intensity radiated field) zračenja
 - Primjeri HIRF-a: mikrovalna pećnica (može ispeći nešto sa zračenjem), radio i radarski tornjevi i uređaji, munje, ...
- Posebno važan i opasan izvor EMI-ja, pogotovo za zrakoplove, su olujni oblaci i **munje**
 - Napon između oblaka i zemlje bude i preko 100 MV, a to znači električno polje jačine oko 100 kV/m i magnetsko polje jačine do 1000 A/m
 - Pri izboju (munji) nastaju ogromne struje (desetci kiloampera) i relativno jako elektromagnetsko zračenje u radio-spektru (HIRF) koje ima poseban „potpis“
 - Zrakoplovi posjeduju posebne antene (ADF) koje iz izboja mogu otkriti smjer i udaljenost oluje

→ Prijenos elektromagnetskih smetnji

– Postoje 4 načina prijenosa EMI-ja:

- 1) **Induktivni** – npr. ako su dva vodiča blizu jedno drugome onda će promjena struje u jednome uzrokovati stvaranje struje u drugome kroz princip elektromagnetske indukcije (promjenjiva struja → promjenjivo magnetsko polje → struja)
- 2) **Konduktivni** – to je prijenos smetnje sa sustava na sustav preko zajedničkog vodiča (npr. kućište ili uzemljenje)
- 3) **Kapacitativni** – ako su dva vodiča jako blizu jedan drugome i na različitom su naponu onda efektivno čine jedan mali kondenzator, a izmjenični signal prolazi kroz kondenzator pa su dakle na taj način „povezani“; to se zove *parazitski kapacitet* i posebno se na njega mora paziti u finoj elektronici poput računala gdje su desetci i stotine tankih vodova na tiskanim pločicama iznimno blizu jedni drugima
- 4) **Radijativni** – to je prijenos elektromagnetskim valovima (npr. HIRF)

→ Zaštita od elektromagnetskih smetnji (EMC – electromagnetic compatibility)

– Postoje tri osnovna principa smanjivanja, odnosno zaštite od EMI-ja:

- 1) **Djelovanje na izvor**, dakle pokušaj sprječavanja nastajanja EMI-ja; neki od mogućih pristupa su:
 - Filteri za uklanjanje naglih šiljaka i neželjenih frekvencija iz signala
 - Omotavanje parova dolaznih i odlaznih žica (kao kod parica i STP-a)
 - Korištenje Faradayevih kaveza (npr. STP za žice, metalne kutije za uređaje, ...)
- 2) **Djelovanje na prijenos**, dakle pokušaj sprječavanja propagacije EMI-ja; neki od mogućih pristupa su:
 - Pravilno grupiranje i razmak vodova (žica) različitih uređaja i sustava
 - Pravilno izoliranje i postavljanje žica prikladnog standard (STP, coax, ...)
 - Pravilno pozicioniranje i razmak između uređaja
- 3) **Djelovanje na žrtvu**, dakle pokušaj sprječavanja primanja/utjecaja EMI-ja; neki od mogućih pristupa su:
 - Korištenje Faradayevih kaveza
 - Pravilno postavljanje i instalacija vodova

– Elektromagnetska kompatibilnost (EMC), odnosno zaštita od elektromagnetske interferencije sve je važnija, osjetljivija i kompliciranija stvar što zrakoplovi koriste više računala i druge sve kompliciranije i osjetljivije elektronike

- Zrakoplovi posebno moraju obraćati pozornost na zaštitu od atmosferskog elektriciteta (uključujući munje)
 - Samim letenjem kroz atmosferu, posebno vlažnu (oblaci, kiša, ...), zrakoplov na trupu nakuplja statički elektricitet
 - Previše statičkog elektriciteta na trupu uzrokovalo bi velika električna polja koja bi mogla utjecati na rad uređaja u zrakoplovu
 - Previše nakupljenog naboja dovelo bi do pražnjenja u atmosferu izbojima (iskre), a što bi uzrokovalo nastanak svakakvih EM valova, dakle radijativnog EMI-ja
 - Na trupu zrakoplova zato postoje posebni štapovi, poput antena, koji su posebno oblikovani kako bi se elektricitet polako i postepeno, bez izboja, oslobađao sa trupa nazad u atmosferu
 - Da bi to funkcioniralo trup zrakoplova mora biti JAKO vodljiv
 - Aluminijski trupovi mogu podnijeti i disipirati jake struje (munje)
 - Kod novijih trupova od karbonskih vlakana morali su povećati i poboljšati vodljivost (vodljivi premazi, metalne mreže, aluminizirana stakloplastika, ...) da bi zrakoplovi bili sigurni
 - Posebno se mora paziti kod popravaka dijelova trupa da je osigurana i vodljivost te da je popravljeni pravilno povezan s ostatkom trupa u vodljivu cjelinu