

# Računarska grafika

Principi prikazivača



# Vektorski prikazivači

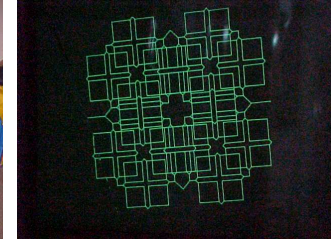
- Vektorski prikazivači (*vector, stroke, line drawing, caligraphic displays*):
  - prikazivački uređaji razvijeni sredinom 60-ih i u širokoj upotrebi do kraja 80-ih
  - termin vektor je korišćen kao sinonim za liniju; *stroke* je kratka linija
  - karakteri su konstruisani kao sekvence takvih linija



Preuzeto sa: <http://excelsior.biosci.ohio-state.edu/~carlson/history/tree/images/sage.jpg>



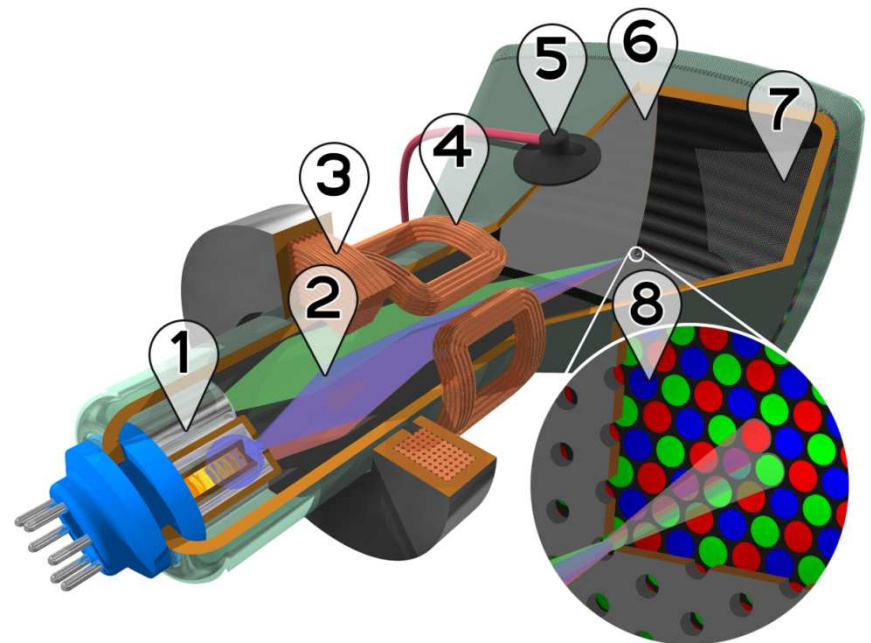
Preuzeto sa: <http://www.cca.org/vector/>



- Tipičan vektorski sistem se sastoji od:
  - displej-kontrolera (DC) povezanog kao U/I jedinica na centralnu procesnu jedinicu (CPU)
  - displej-bafer memorije
  - katodne cevi (CRT)

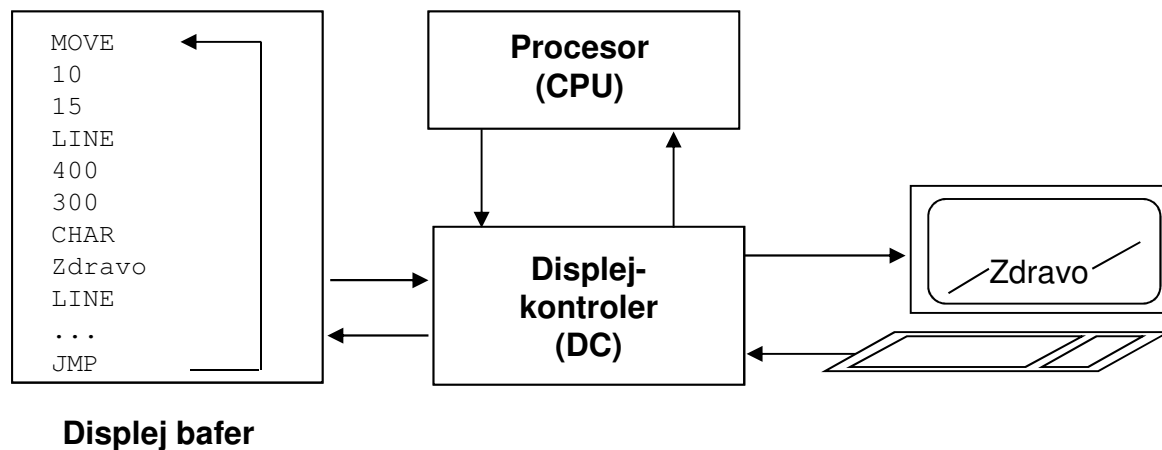
# Katodna cev

- Katodna cev (CRT)
  - 1. elektronski topovi
  - 2. elektronski mlazevi
  - 3. kalemovi za fokusiranje
  - 4. kalemovi za odklon
  - 5. spoj anode
  - 6. maska
  - 7. fosfor na staklu cevi
  - 8. trijade RGB fosfora



Preuzeto sa: [http://en.wikipedia.org/wiki/Image:CRT\\_color\\_enhanced.png](http://en.wikipedia.org/wiki/Image:CRT_color_enhanced.png)  
(Created by Søren Peo Pedersen)

# Organizacija vektorskog prikazivača



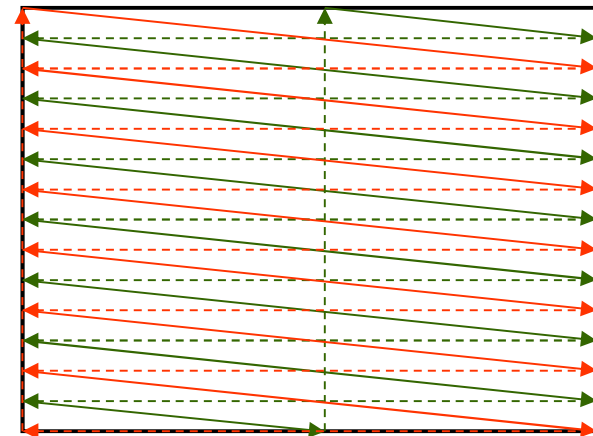
- Bafer sadrži displej-program koji je generisao računar
- Displej-program sadrži komande za:
  - pomeranje “pera” u tačku (MOVE x,y)
  - crtanje linija od tekuće tačke do zadate tačke (LINE x,y)
  - crtanje karaktera (CHAR string)
- Komande za crtanje interpretira displej-kontroler
- Kontroler šalje digitalne koordinate tačka generatoru vektora

# Rad vektorskog prikazivača

- Generator vektora konvertuje koordinate tačaka u analogne napone kola za otklon mlaza
- Kolo za otklon pomera elektronski mlaz koji „piše“ po fosfornoj oblozi CRT
- Suština je da zrak prati tačku po tačku specificiranim redosledom displej-programa
- Tehnika prikaza na vektorskim uređajima se naziva
  - slučajnim skeniranjem (*random-scan*)
- Pošto svetlosni izlaz fosfora opada u desetinama ili najviše stotinama  $\mu\text{s}$ , DC mora ciklično da izvršava program da osveži fosfor barem sa 30Hz
- Bafer koji sadrži program - bafer za osvežavanje (*refresh buffer*)
- Instrukcija JMP na kraju programa formira petlju za ciklično osvežavanje
- Šezdesetih 20. veka, bafer memorija i dovoljno brzi kontroleri da osvežavaju na (barem) 30Hz bili su skupi
  - samo nekoliko hiljada linija se moglo prikazati bez vidnog treperenja

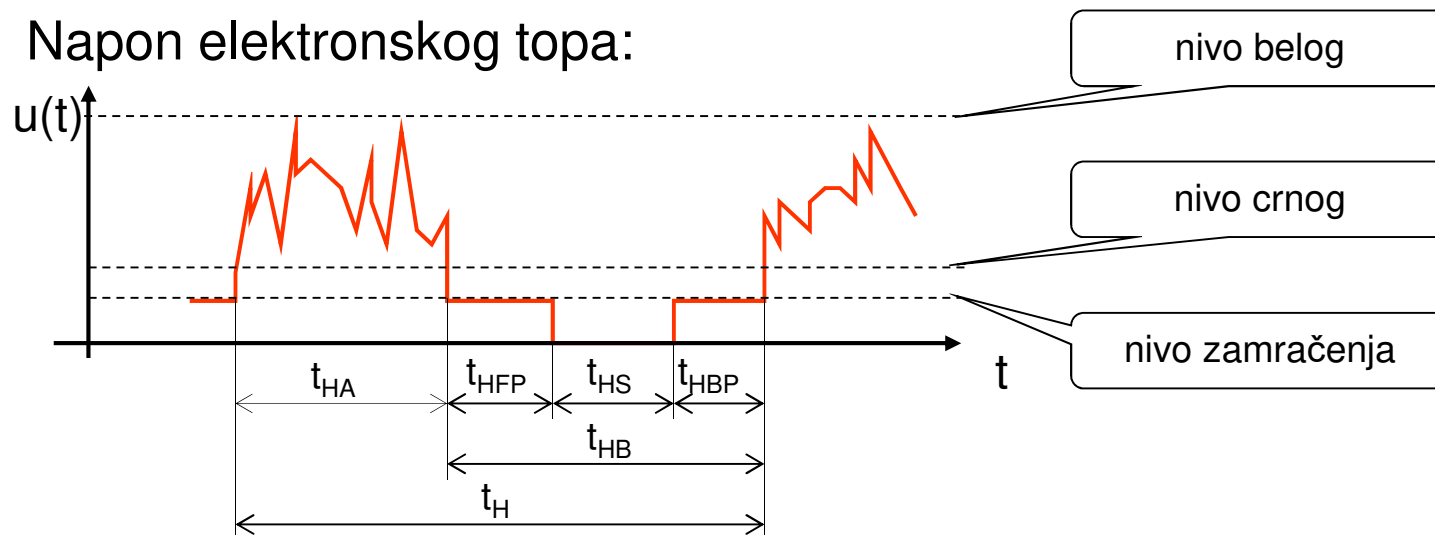
# Rasterski prikazivači - TV

- U ranim sedamdesetim godinama počinje razvoj rasterske grafike
- Smatra se da je rasterska grafika doprinela razvoju oblasti više nego bilo koja druga tehnologija
- Rasterska grafika je zasnovana na tehnologiji televizije (*raster-scan*)
- Pre televizije – film
- Perzistencija ljudskog vida:
  - najviše 24 različite slike u sekundi
- Princip filma (pokretne slike):
  - osvetljena slika, zamračenje,...
- Iz tehničkih razloga u TV:
  - 25-30 slika/s
- Za vreme  $1/30s$  osećaj osvetljaja jenjava – treperenje
- Ideja – dva puta osvetliti ekran
  - jedna slika iz 2 poluslike (*interlacing*)



# TV signal

- Napon elektronskog topa:



- Horizontalna perioda mlaza:  $t_H = t_{HA} + t_{HB} = t_{HA} + (t_{HFP} + t_{HS} + t_{HBP})$
- Horizontalna frekvencija (učestanost linije):  $f_H = 1/t_H$
- Po analogiji, perioda slike:  $t_V = t_{VA} + t_{VB} = t_{VA} + (t_{VFP} + t_{VS} + t_{VBP})$
- Vertikalna frekvencija (učestanost slike):  $f_V = 1/t_V$

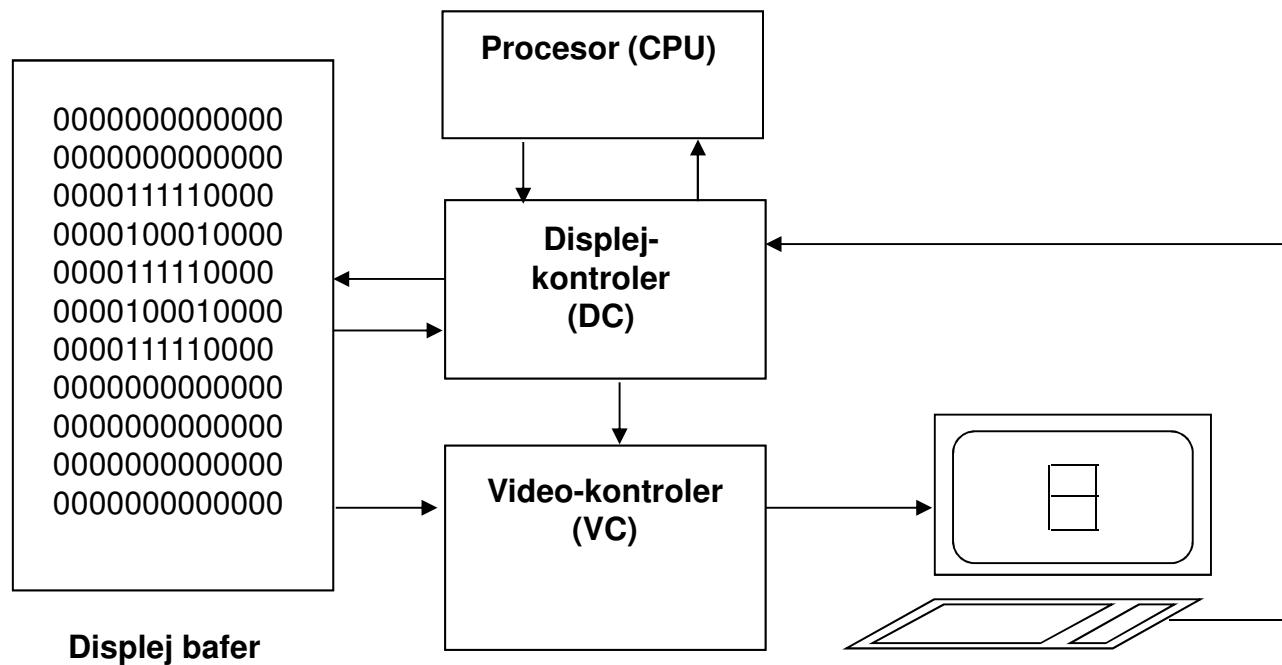
# Rasterski prikazivači

- Rasterski prikazivači (*raster display*)
  - smeštaju primitive za prikaz u bafer za osvežavanje (video-memoriju)
  - primitive su linije (prave i lukovi), karakteri i popunjene površi
  - primitive su razložene na piksele, kao komponente od kojih su obrazovane
- Displej-kontroler prima i interpretira sekvence komandi
  - komande su slične onima kod vektorskih displeja
    - definišu primitive u vektorskom obliku
  - kontroler vrši rasterizaciju – konverziju iz vektorskog u rasterski oblik
- U jednostavnijim sistemima, kao što su bili kod ranih personalnih računara:
  - kontroler displeja postoji samo kao softverska komponenta grafičkog paketa
  - bafer za osvežavanje je samo deo CPU memorije
  - iz bafera može da čita podsistem za prikaz slike
    - često se naziva video-kontrolerom
  - video-kontroler proizvodi stvarnu sliku na ekranu



# Organizacija rasterskog prikazivača

- Binarno-monohromatska slika 8 u displej-baferu



# Raster

- Kompletna slika na rasterskom prikazivaču se formira iz rastera
- Raster predstavlja matricu:
  - niz horizontalnih rasterskih linija
  - svaka linija je red individualnih piksela
- Raster se čuva kao matrica piksela koja reprezentuje celu površ ekrana
- Cela slika se skenira sekvencijalno od strane video-kontrolera:
  - linija po linija od vrha prema dnu i zatim ponovo od vrha
  - za svaki piksel, intenzitet mlaza se postavlja tako da reflektuje intenzitet piksela
  - u sistemima sa bojom kontrolišu se tri mlaza
    - po jedan za crvenu, zelenu i plavu boju prema odgovarajućim komponentama vrednosti svakog piksela

# Bafer za osvežavanje slike

- Frekvencija osvežavanja slike:
  - u ranim danima rasterske grafike – TV učestanost od 25-30Hz, sa protkivanjem
  - danas - od 60Hz naviše (120Hz za stereoviziju), bez protkivanja
- Sadržaj bafera za osvežavanje slike:
  - u vektorskim sistemima: program (kodovi operacija, koordinate krajnjih tačaka)
  - u rasterskim sistemima: pikseli slike
- Termini:
  - bafer slike (bafer za osvežavanje) – memorija u kojoj se čuva slika
  - “bitmapa” i “piksmapa” – koriste se za označavanje oba pojma:
    - bafer slike (memorija) i
    - niz vrednosti piksela koje se mapiraju 1:1 u piksele na ekranu (sadržaj memorije)
    - termin “bitmapa” – uglavnom se koristi za binarno-monohromatsku sliku
    - termin “piksmapa” (*pixmap*, skraćunica za *pixel map*) - za sisteme sa više bita po pikselu

# Memorija rasterskih prikazivača

- Memorija bafera slike:
  - jeftina poluprovodnička RAM za bitmape se pojavila u ranim 70-im
    - proboj koji je učinio da rasterska grafika postane dominantna tehnologija
  - najpre – jedan bit po pikselu, dvonivoska (binarno-monohromatska) slika
    - bitmapa za ekran sa rezolucijom 1024 x 1024 piksela je samo 128KByte
  - kasnije – više bita po pikselu, slike u skali sivog
    - 8 bita po pikselu, 256 nijansi sive, za 1024 x 1024 piksela – 1MByte
  - danas – barem 8 bita po osnovnoj (R,G,B) boji piksela, slika u boji
    - 16 miliona boja (*true color*), za 1024x1024 – 3MByte

# Rasterski prikazivači u boji

- Rani sistemi u boji:
  - 8 bita po pikselu, dopuštaju 256 boja (u isto vreme na ekranu)
- Kasniji sistemi:
  - 24 bita po pikselu, dopuštaju 16 miliona boja
  - 24 bita i rezolucija 1280 x 1024 zahteva samo 3.75MB RAM-a
  - 32 bita po pikselu
    - 24 bita su namenjena za reprezentaciju boje
    - 8 u kontrolne svrhe (alfa kanal)
- Savremeni sistemi:
  - 96 i više bita po pikselu na rezoluciji 1920 x 1080 (ili većoj):
    - 2x32 bita se koristi za dvostruko baferisanje – jedna slika se prikazuje, druga crta
    - 4x32 bita se koristi za četverostruko baferisanje – kod stereoskopije
    - 24 bita se koristi za Z-bafer
    - 8 bita za *stencil*-bafer

# Prednosti rasterske grafike

- Glavne prednosti rasterske grafike nad vektorskom:
  - niža cena, skenirajuća logika jednostavnija
    - regularno, repetitivno rastersko skeniranje je daleko brže i manje skupo za implementaciju od slučajnog skeniranja vektorskih sistema
    - generatori vektora moraju da budu veoma precizni da obezbede linearnost i ponovljivost otklona mlaza
  - sposobnost prikazivanja popunjene površi bojom ili teksturom
    - bogat način prenosa informacije, važan za realistične slike 3D objekata
  - proces osvežavanja je nezavisan od kompleksnosti slike (broja primitiva, npr. poligona):
    - većina ne zapaža treperenje na rasterskim ekranima sa  $f > 70\text{Hz}$
    - vektorski prikazivači trepere kada broj primitiva u baferu postane preveliki

# Nedostaci rasterske grafike (1)

- Obaveza rasterizacije (*sken-konverzije*):
  - primitive kao što su linije i poligoni se zadaju pomoću njihovih temena i moraju da se konvertuju u piksele koji predstavljaju primitive u baferu slike
  - rasterizacija može da se obavlja:
    - softverski (gde je CPU odgovoran za svu grafiku)
    - specijalno-namenskim hardverom - čipom procesora rasterske slike (*raster image processor, RIP*) korišćenim kao koprocesor ili akcelerator
- Dinamika za realno-vreme je računski daleko zahtevnija na rasterskim sistemima:
  - transformisanje 1000 nepovezanih linija na vektorskom sistemu
    - transformisanje 2000 njihovih krajnjih tačaka
  - u rasterskom sistemu se transformišu krajnje tačke + svaka linija mora tada da bude rasterizovana koristeći njene nove krajnje tačke
  - kada je CPU odgovoran i za transformaciju krajnjih tačaka i za rasterizaciju, samo mali broj primitiva može da bude transformisan u realnom vremenu
  - HW za transformacije i rasterizaciju je potreban zbog dinamike u rasterskim sist.
  - kao rezultat napretka u VLSI, ovo je postalo ostvarivo čak u skromnijim sistemima

# Nedostaci rasterske grafike (2)

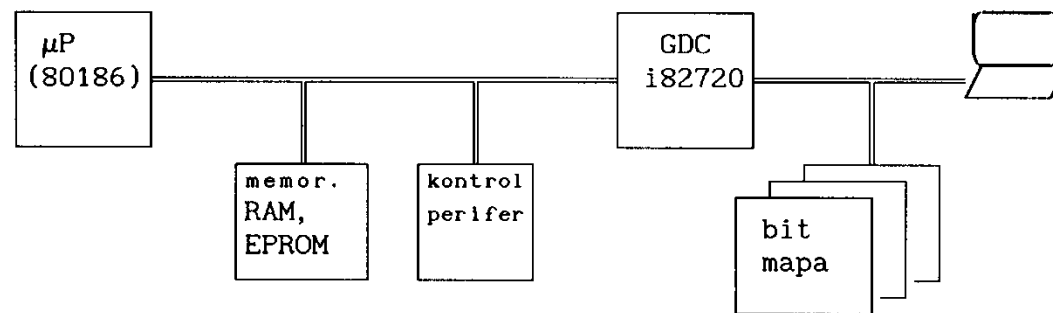
- Aproksimativna priroda prikaza glatkih linija:
  - potiče od same prirode rastera
  - vektorski sistem može da crta kontinualnu, glatku liniju (i čak neke glatke krive)
  - rasterski sistem može samo da aproksimira glatke linije pikselima na rasterskoj mreži
    - ovo može da prouzrokuje poznati problem "nazupčenosti"
    - u obradi signala ovakva manifestacija (greške uzorkovanja) se naziva *aliasing*
    - pojavljuje se kada se funkcija kontinualne promenljive sa naglim promenama intenziteta aproksimira diskretnim uzorcima
  - tehnike za *antialiasing* na sistemima sa skalom sivog ili u boji
    - gradacija u intenzitetu susednih piksela na ivicama primitiva umesto isključivog postavljanja piksela na maksimalni ili nulti intenzitet





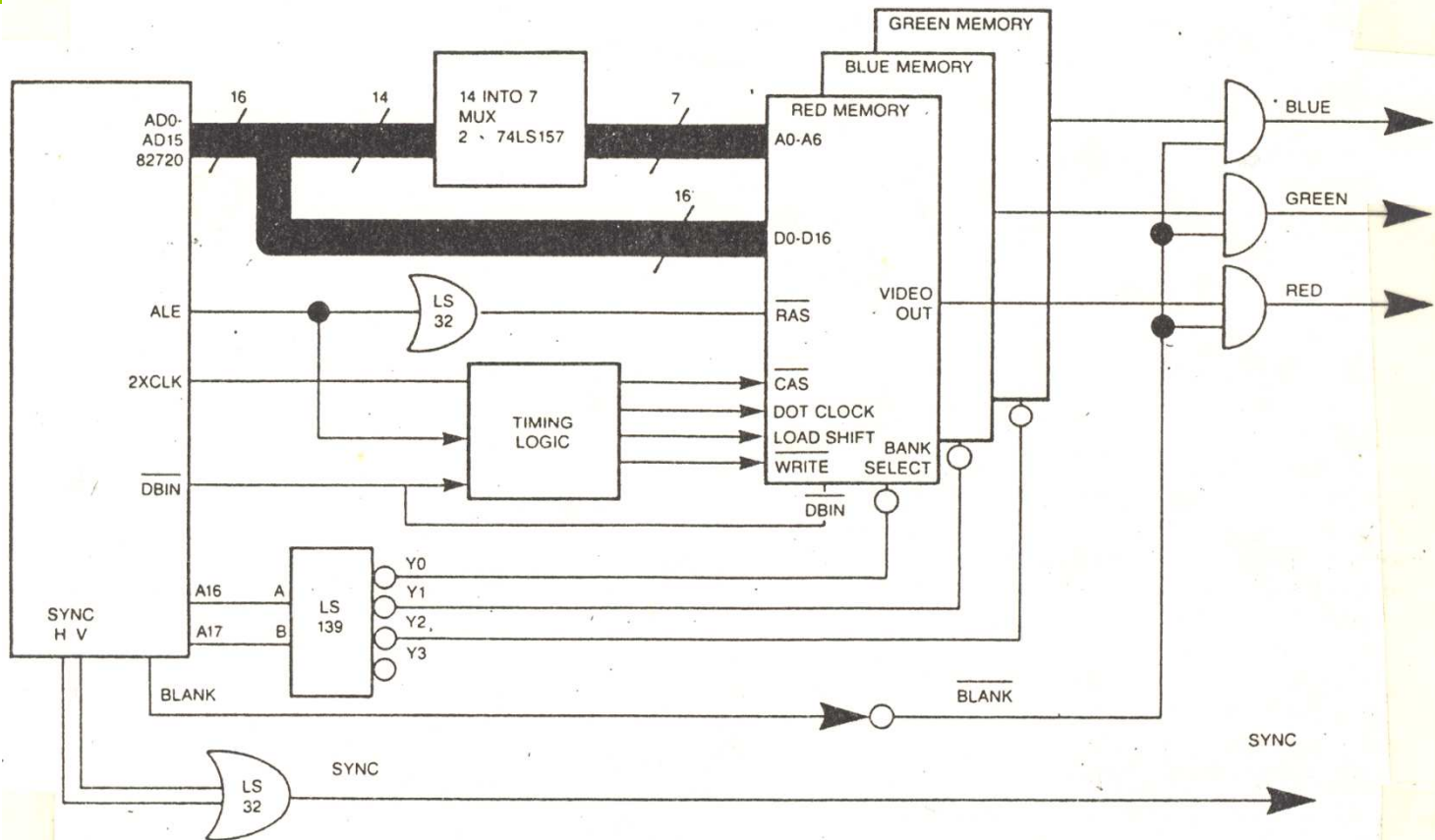
# Princip grafičkog kontrolera

- Primer – grafički displej-kontroler i82720



- Projektovanje grafičkog kontrolera
  - rezolucija: 512x512
  - broj boja: 8
  - pixmapa:  $\log_2 8 = 3 \text{ bit/pix}$ 
    - 3 memorijske “ravni” sa 1bit/pix, svaka za jednu osnovu boju RGB

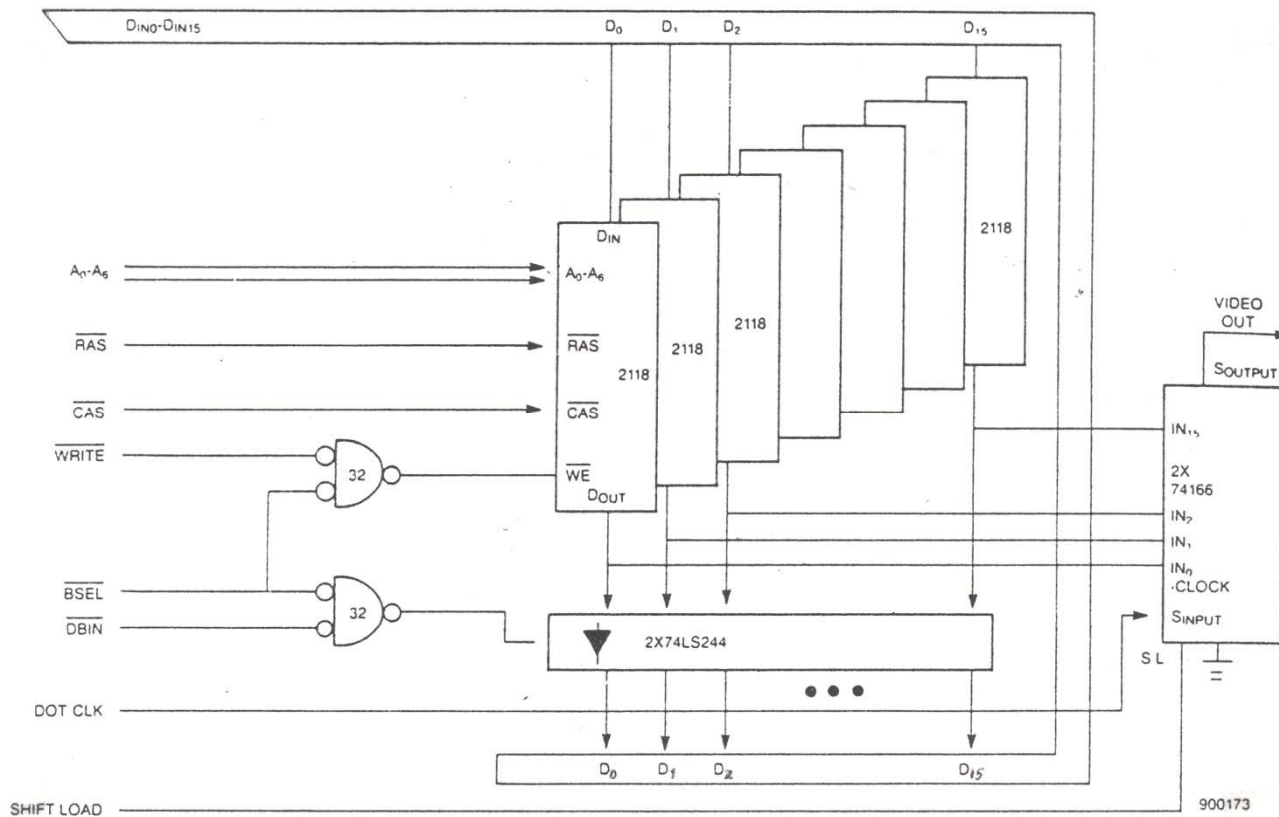
# Organizacija video-memorije



# Proračun video memorije

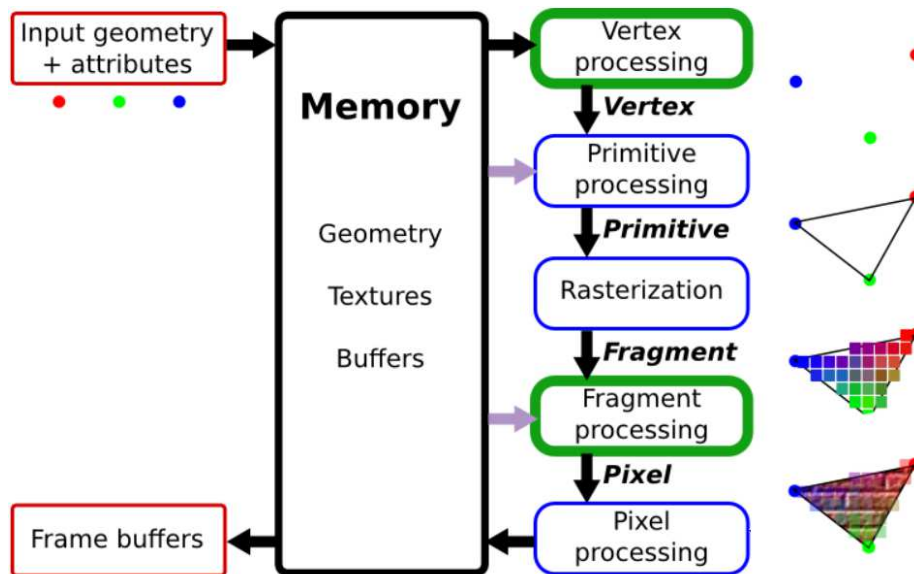
- Za rezoluciju  $512 \times 512 = 256\text{Kpixel} \times 1 \text{ bit/pixel} = 256\text{Kbit}$ 
  - organizacija memorijske banke:  $16\text{Kword} \times 16\text{bit/word} = 256\text{Kbit}$
  - čip dinamičkog RAM-a 2118:  $16\text{K} \times 1\text{bit}$ 
    - potrebno 16 čipova, svaki za 1 bit reči od 16bita
    - za adresiranje  $16\text{Kbita} = 2^{14} \text{ bita} = 2^7\text{row} \times 2^7\text{col}$ 
      - po 7 adresnih linija za adresu reda, odnosno kolone
- Signali RAS i CAS (*row/column address strobe*) sa odgovarajućom adresom aktiviraju vrstu/kolonu memorijskog čipa
- BSEL selektuje banku memorije
- DBIN se aktivira za čitanje iz video memorije

# Organizacija jedne banke VM

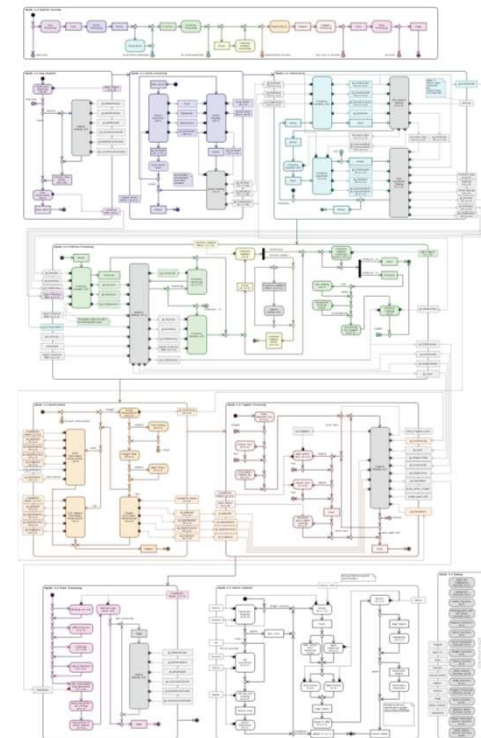


# Grafička protočna obrada (1)

- OpenGL 4.4 – šematski prikaz:
- Pojednostavljeno (OpenGL 2):



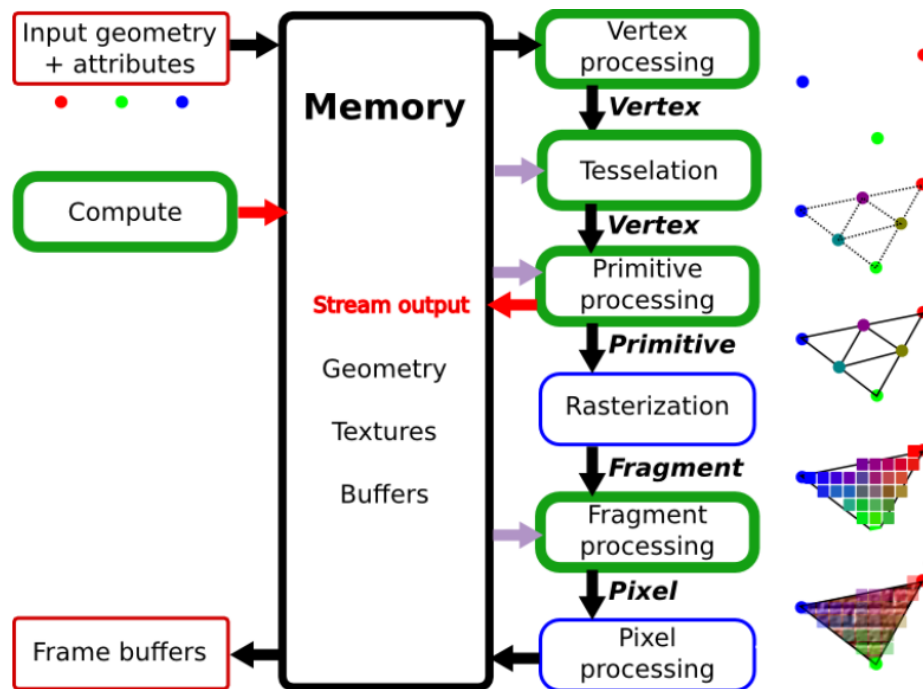
Preuzeto sa: <http://romain.vergne.free.fr/teaching/IS/SI03-pipeline.html>



Preuzeto sa: <http://openglinsights.com/pipeline.html>

# Grafička protočna obrada (2)

- OpenGL 4 – novi tipovi programa za senčenje (*shaders*)



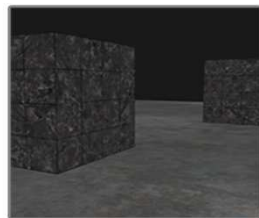
Preuzeto sa: <http://romain.vergne.free.fr/teaching/IS/SI03-pipeline.html>

# Memorija grafičkog kontrolera

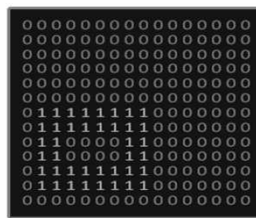
- Baferi:
  - bafer boje (*color/screen buffer*)
  - bafer dubine (*depth/Z buffer*)
  - bafer šablona (*stencil buffer*)
- Memorija za geometriju (temena, stranice, normale)
- Memorija za teksturu

# Bafer šablona

- Bafer šablona (*stencil buffer*)
  - obično sadrži 8 bita po pikselu (256 različitih vrednosti)
  - obično deli istu memoriju sa z-baferom, npr. 24:8 bita, *depth:stencil*
- Test šablona (*stencil test*) prihvata/odbacuje crtanje piksela
  - poredi se sadržaj bafera šablona sa zadatom referentnom vrednošću
  - sledi obradu fragmenta, prethodi testu dubine
- Sadržaj bafera šablona može da se ažurira za vreme iscrtavanja da se postignu neki efekti



bafer boje



bafer šablona



rezultantna slika

Preuzeto sa:  
<https://learnopengl.com/>  
#!Advanced-OpenGL/Stencil-testing



# Bafer šablona – korišćenje

- Jednostavan način korišćenja bafera šablona
  - omogući se upis u bafer šablona
  - crtaju se objekti, ažurirajući sadržaj bafera šablona
  - onemogući se upis u bafer šablona
  - crtaju se objekti uz omogućen test šablona
- OpenGL / Direct3D imaju podršku za rad sa baferom šablona
- Funkcija testa šablona:
  - primenjuje se na vrednost u baferu i zadatu referentnu vrednost
  - NEVER, ALWAYS, EQUAL, NOTEQUAL, LESS, LEQUAL, GREATER, GEQUAL

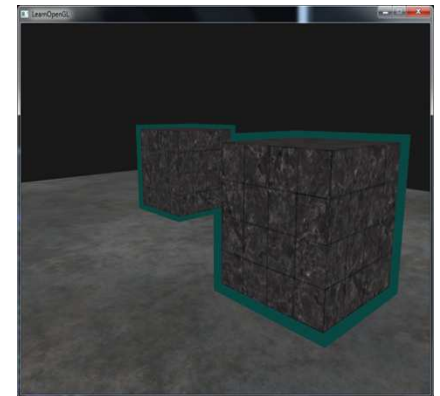
# Bafer šablona – operacija za upis

- Operacija za upis u bafer šablona ima tri parametra
  - opisuju šta se radi u sledećim slučajevima:
    - test šablona nije prošao
    - test šablona prošao, nije prošao test dubine
    - oba testa prošla
  - moguće akcije koje određuju šta se radi u odgovarajućem slučaju:

Akcija	Opis
GL_KEEP	Zadržava se tekuća vrednost smeštena u bafer
GL_ZERO	Vrednost se postavlja na 0
GL_REPLACE	Vrednost se zamenjuje referentnom vrednošću
GL_INCR	Vrednost se inkrementira za 1 ako je manja od maksimalne
GL_INCR_WRAP	Isto kao GL_INCR, samo postaje 0 ako se prevaziđe maksimum
GL_DECR	Vrednost se dekrementira za jedan ako je veća od minimalne
GL_DECR_WRAP	Isto kao GL_DECR, samo postaje maksimalna vrednost ako podbaci 0
GL_INVERT	Invertuje bite tekuće vrednosti u baferu šablona

# Bafer šablona – primena

- Rastersko oivičenje
  - postavi se `ALWAYS` funkcija i ref. vr. 1
  - iscrtaju se objekti
    - u baferu šablona je 1 na mestu gde su objekti vidljivi
  - postavi se `NOTEQUAL` funkcija i ref. vr. 1
    - jedinice u baferu šablona štite nacrtane objekte
  - onemogućí se test dubine i upis u bafer šablona
  - skalira se svaki od objekata malim skala faktorom
  - nacrtaju se objekti bojom oivičenja
  - omogućí se upis u bafer šablona i test dubine



Preuzeto sa:  
<https://learnopengl.com/>  
#!Advanced-OpenGL/Stencil-testing

# Karakterisitike kontrolera – primer

- NVIDIA Geforce RTX 4090: DLSS3 (*Deep Learning Super Sampling*)
  - CUDA jezgara (*cores*): 16,384
  - jezgra za senčenje (*shading*): arhitektura *Ada Lovelace*, 83 TFLOPS
  - jezgra za praćenje zraka (*ray tracing*): 3. gen., 191 TFLOPS
  - jezgra za tenzore (AI) (*tensor*): 4. generacija, 1321 AI TOPS
  - osnovni takt (*base clock*): 2.23GHz
  - takt pod opterećenjem (*boost clock*): 2.52GHz
  - kapacitet i tip video memorije: 24GB GDDR6X
  - širina memorijskog interfejsa (*memory interface width*): 384-bit
  - rezolucija i učestanost: 4K na 240Hz ili 8K na 60Hz sa DSC i HDR
  - cena: ~\$2,100.00

# Savremeni grafički kontroleri

AMD Radeon  
RX 7900 XTX



By Geni - Photo by User:geni, CC BY-SA 4.0,  
<https://commons.wikimedia.org/w/index.php?curid=128619096>



<https://www.amazon.com/PNY-VCNRTXA6000-PB-NVIDIA-RTX-A6000/dp/B09BDH8VZV>



nVIDIA GeForce RTX 4090

nVIDIA RTX A6000

<https://www.amazon.com/NVIDIA-GeForce-Founders-Graphics-GDDR6X/dp/B0BJFRT43X>