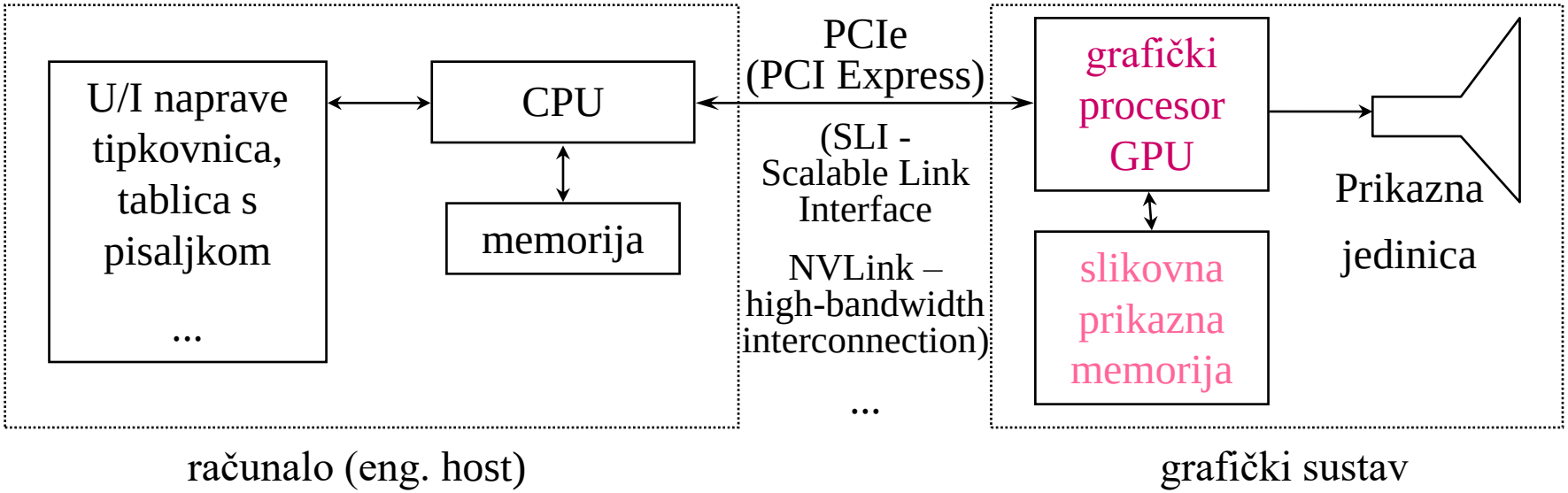
2 Računalna grafička oprema

- sklopovska grafička oprema
  - grafički procesor GPU
    - rasterska prikazna procesna jedinica
    - vektorska prikazna procesna jedinica
  - ulazne grafičke naprave
  - izlazne grafičke naprave

- programska grafička oprema
  - knjižnica grafičkih rutina
  - grafička jezgra načinjena u okviru standarda (API), jezici za sjenčanje
  - gotovi programski paketi
    - za crtanje - CAD, animacije
    - za prikaz podataka

2.1 SKLOPOVSKA GRAFIČKA OPREMA

Povezanost grafičkog procesora s ostalim jedinicama sustava

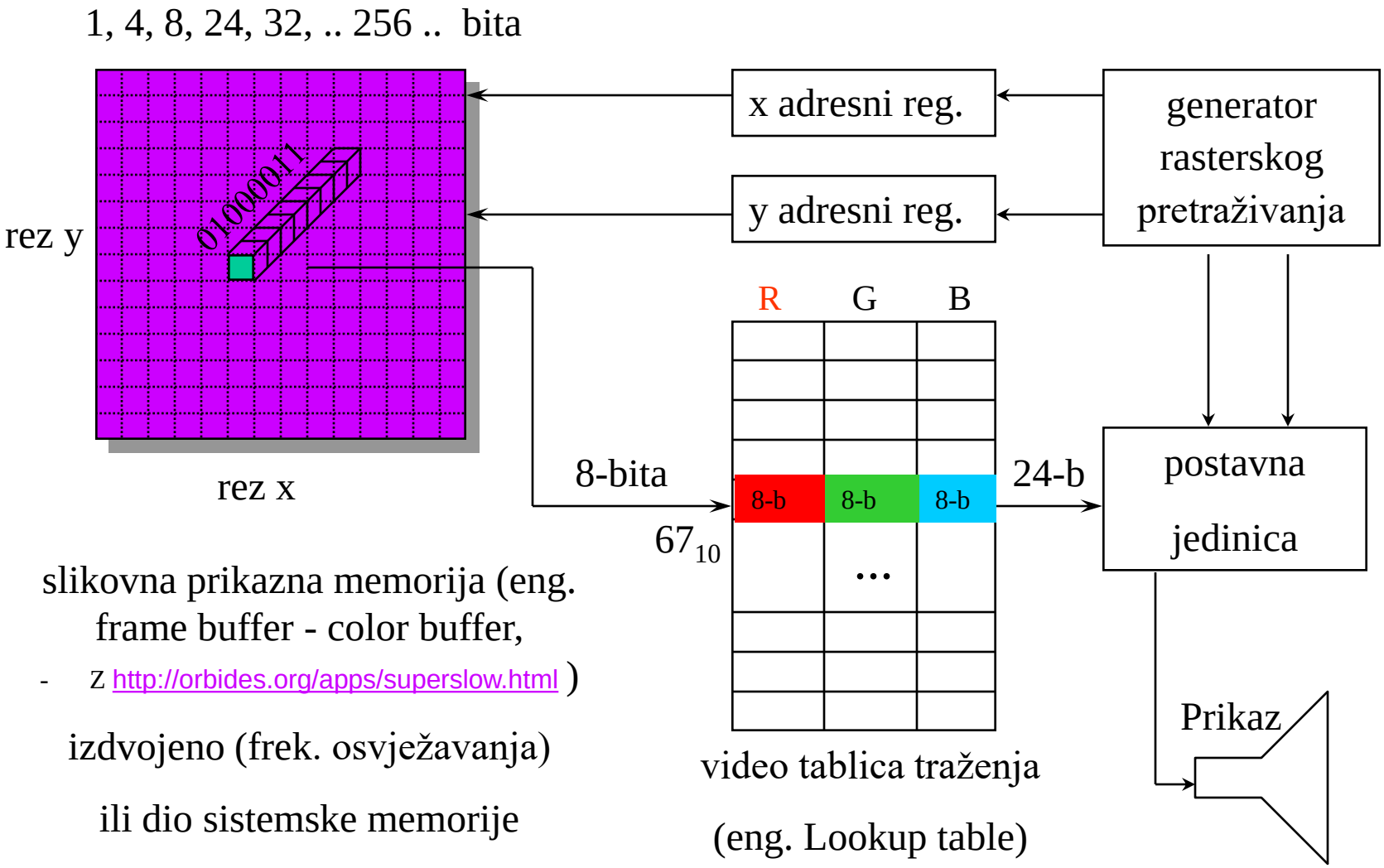


* Primjer sklopovske grafičke opreme

Ž. M,

2.1.1 Grafički procesor GPU

Funkcija rasterske prikazne procesne jedinice

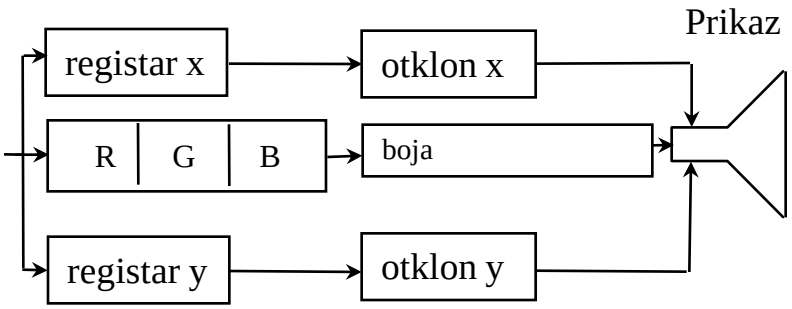


slikovna prikazna memorija (eng. frame buffer - color buffer,

- z http://orbides.org/apps/superslow.html)

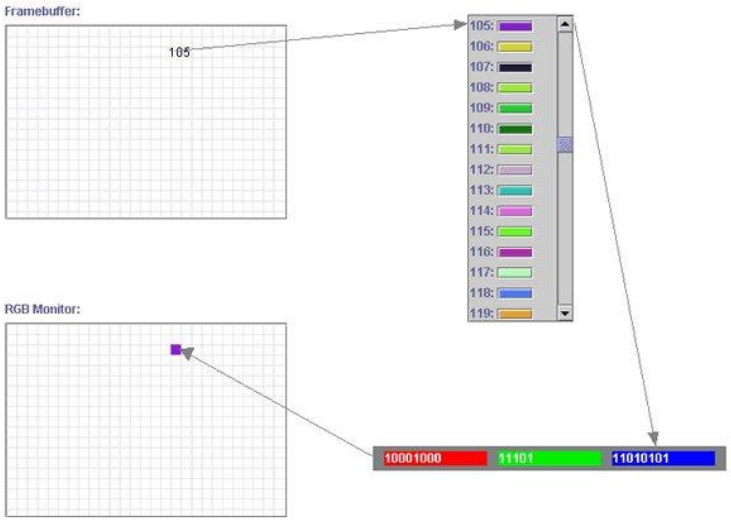
izdvojeno (frek. osvježavanja)

ili dio sistemske memorije



postavna jedinica

LUT tablica



http://threejs.org/examples/webgl_geometry_colors_lookuptable.html (a)

1-bit
2 boje

4-bita (LUT)
16 boja

8-bita (LUT)
256 boja

24-bita
16 777 216 boja

30/36/48-bit
Deep color

http://app.meemoo.org/#gist/3721129

Povijesni razvoj GPU-a

- profesionalno grafičko sklopovlje - razvoj zadnjih ~ 35 godina
- osobna računala

’91 tvrtka S3 kartica, ViRGE, no naprednije mogućnosti spore

’96 tvrtka 3DFX kartica Voodoo, 3D ubrzivačka kartica (nema 2D)

do ’99 sklopovski implementirane funkcije – postiže se velika brzina, no programirljivo sklopovlje (CPU) je fleksibilnije (ovisno o problemu u nekim slučajevima može biti brže)

’99 važne grafičke funkcije sklopovski su podržane – GPU

’01 kartica GeForce3 podržava male **programe** u geometrijskoj fazi
vrlo mali, jednostavne aritmetičke operacije (engl. vertex shader)

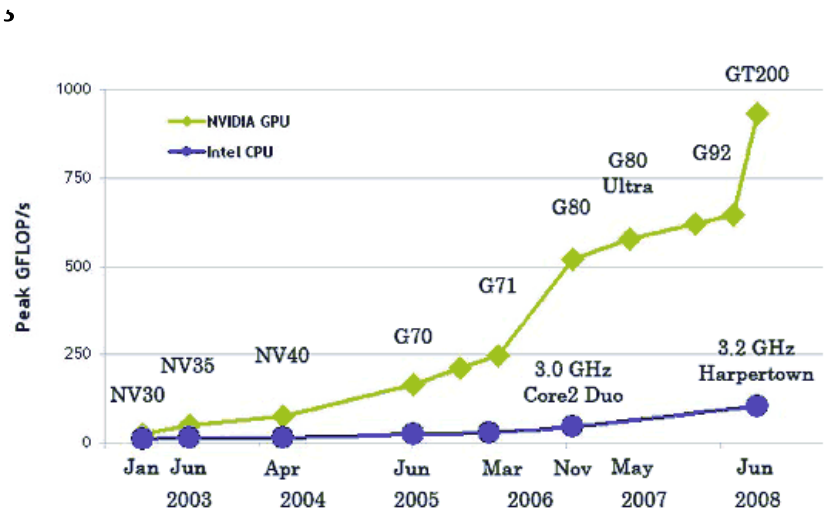
’02 programi za sjenčanje slikovnih elemenata, **floating point**
dodaje se pristup teksturama (engl. pixel shader, fragment shader)
još uvijek nema prave kontrole toka, postoje uvjetne naredbe ADDNZ ali ne i naredbe skoka JMP

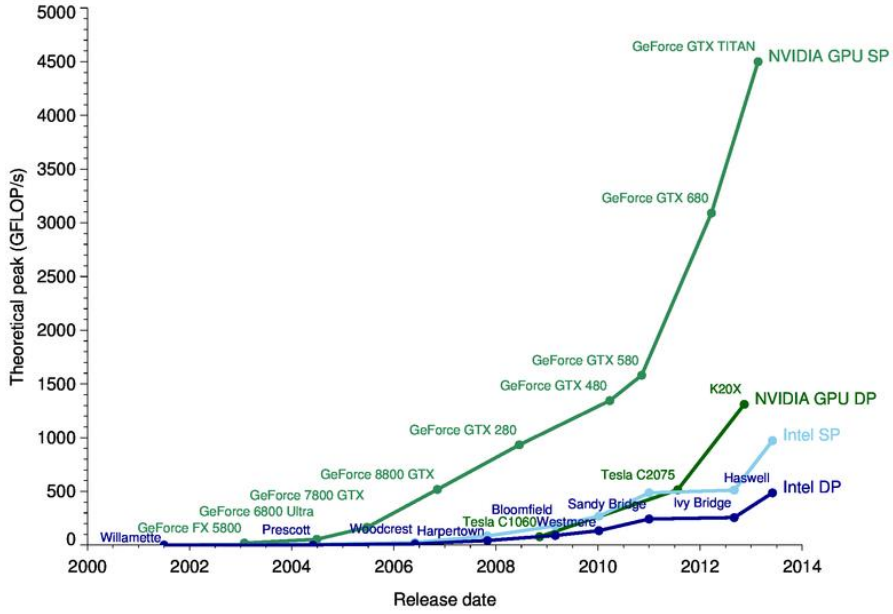
’04 kartica GeForce6800 **kontrola toka** – naredbe skoka
povećavanje broja cjevovoda

’06 GPGPU (*General-Purpose computation on GPU*) masivno paralelne arhitekture npr. Tesla (Nvidia) programska podrška CUDA

Povijesni razvoj GPU-a – primjer

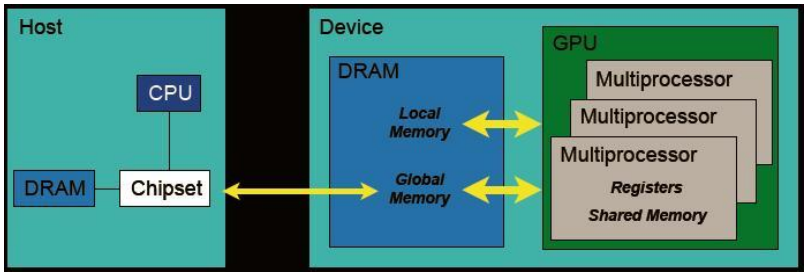
Generation	Year	Product Name	Process	Transistors	Antialiasing Fill Rate	Polygon Rate	Note
First	Late 1998	RIVA TNT	0.25 µ	7 M	*50 M*	6 M	1
First	Early 1999	RIVA TNT2	0.22 µ	9 M	*75 M*	9 M	2
Second	Late 1999	GeForce 256	0.22 µ	23 M	*120 M*	15 M	3
Second	Early 2000	GeForce2	0.18 µ	25 M	*200 M*	25 M	4
Third	Early 2001	GeForce3	0.15 µ	57 M	800 M	30 M	5
Third	Early 2002	GeForce4 Ti	0.15 µ	63 M	1200 M	60 M	6
Fourth	Early 2003	GeForce FX	0.13 µ	125 M	2000 M	200 M	7





Npr: Grafički procesori za mobitele - mali (ARM)

GPGPU (General-Purpose computation on GPU)



- veliki broj procesnih jedinica SIMD (*manycore chip*) procesori toka (*stream processors*)
- podatkovno paralelna superračunala
- iznimna računalna moć TFLOPs – 10^{12} (teraflops) (Peta FLOPs) (*multithread*)
- npr. Nvidia Titan RTX, 130 Tensor TFLOPS, 576 Tensor Cores, 4608 Cuda, 72 RT Cores, 2500$, 2019

Primjena (BIG DATA)

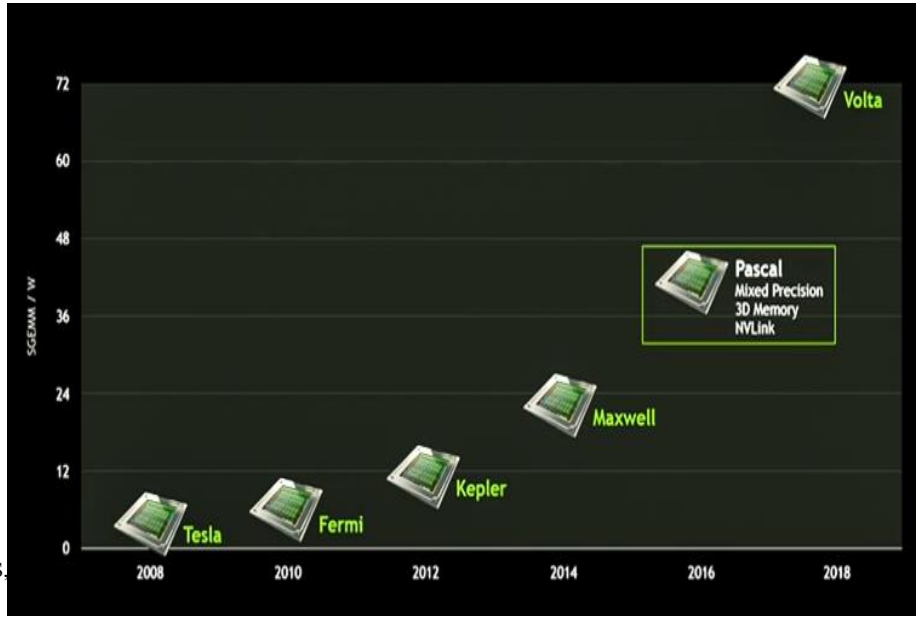
- bioinformatika, genetika, transport, AI,
- fizikalni modeli - dinamika fluida, astrofizika, dinamika molekula

Npr. Sklopovski model, programska sučelja – API

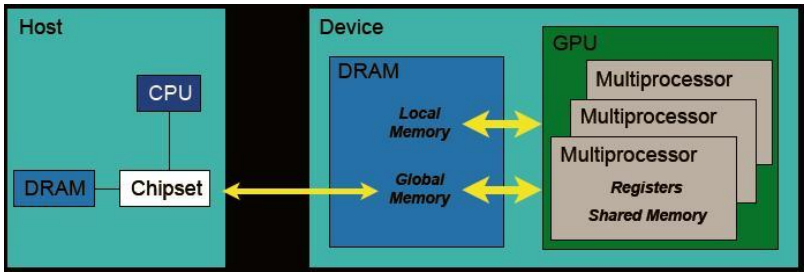
– CUDA (*Compute Unified Device Architecture*) Nvidia

- CTM (*Close To Metal*) ATI (AMD)
- OpenCL (*Open Computing Language*) Apple 2008, Khronos
- Vulkan

Turing



GPGPU (General-Purpose computation on GPU)



Programski model – CUDA

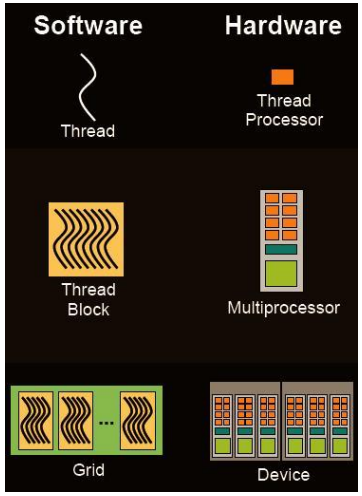
- skalabilni model za paralelno programiranje (programira se iz C-a s CUDA alatima)
  - serijski dio koda izvodi se na CPU
  - paralelni dio na GPU

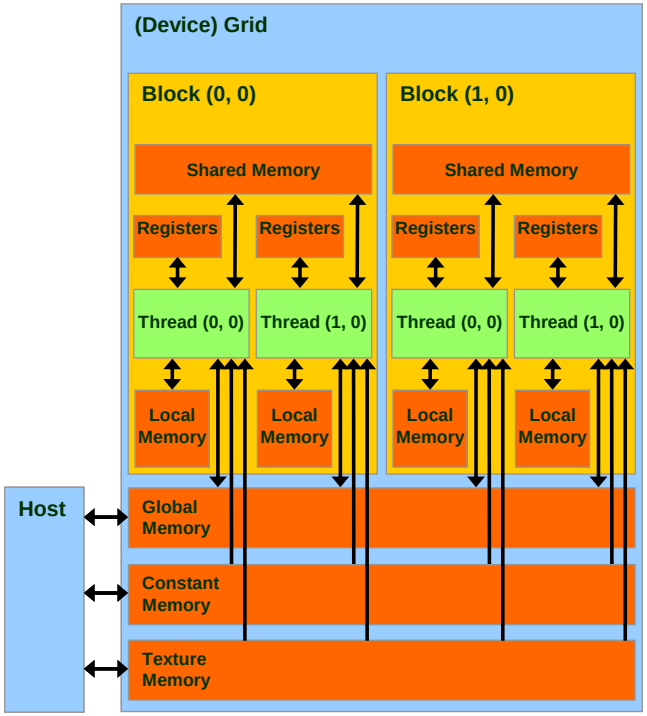
- definira na logičkoj razini **Grid** koji se sastoji od **Blokova** koji su niz dretvi (**Thread**) što se preslikava na fizičku razinu **Uređaja** koji ima **Multiprocesore** koji se sastoje od **Procesnih elemenata**

Praćenje parametara grafičke kartice https://www.techpowerup.com/gpuz/

CUDA biblioteke

- BLAS (*Basic Linear Algebra Subprograms*)
- FFT





Slikovna prikazna memorija (eng. frame buffer)

– memorija u koju se pohranjuje slika,
iz te memorije se obavlja osvježavanje na zaslonu

- pohranjivanje slike GL_COLOR_BUFFER
- udaljenost od očišta GL_DEPTH_BUFFER,

  Z- fighting https://www.realtimerendering.com/erich/udacity/exercises/unit2_zfighting_exercise.html
  logaritamske vrijednosti u Z-spremniku http://threejs.org/examples/#webgl_camera_logarithmicdepthbuffer
- dvostruki spremnik GL_DOUBLE_BUFFER, GL_STEREO
- spremnik maske GL_STENCIL_BUFFER
- kombiniranje slike iz niza slika GL_ACCUM_BUFFER

  GL_AUX_BUFFERS

– spremnik teksture GL_TEXTURE_1D

- GL_TEXTURE_2D
- GL_TEXTURE_3D
- (6 tekstura na kocki) GL_TEXTURE_CUBE_MAP

  http://math.hws.edu/graphicsbook/source/threejs/skybox.html http://math.hws.edu/graphicsbook/demos/c7/cube-camera-demo.html
  https://threejs.org/examples/webgl_panorama_cube https://threejs.org/examples/webgl_materials_cubemap

  određivanje broja bita u spremniku – različite dubine spremnika
- sklopovska podrška za neke funkcionalnosti spremnika – brzo – npr. brisanje, logičke operacije, operacije usporedbe, akumulacije/stapanja, antialias

Upotreba spremnika – OpenGL

glutInitDisplayMode (GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH);

- inicijalizacija spremnika

  - GLUT_RGB - spremnik boje
  - GLUT_DOUBLE - dvostruki spremnik i osvježava se sa glutSwapBuffers();

  inače ako ne koristimo dvostruki spremnik imamo glFlush();

  - GLUT_DEPTH - spremnik udaljenosti i nakon stvaranja prozora

  window = glutCreateWindow ("Prozor");

  potrebno je omogućiti spremnik

  ```
  glEnable(GL_DEPTH_TEST);
  glDepthFunc(GL_LEQUAL)
  ```

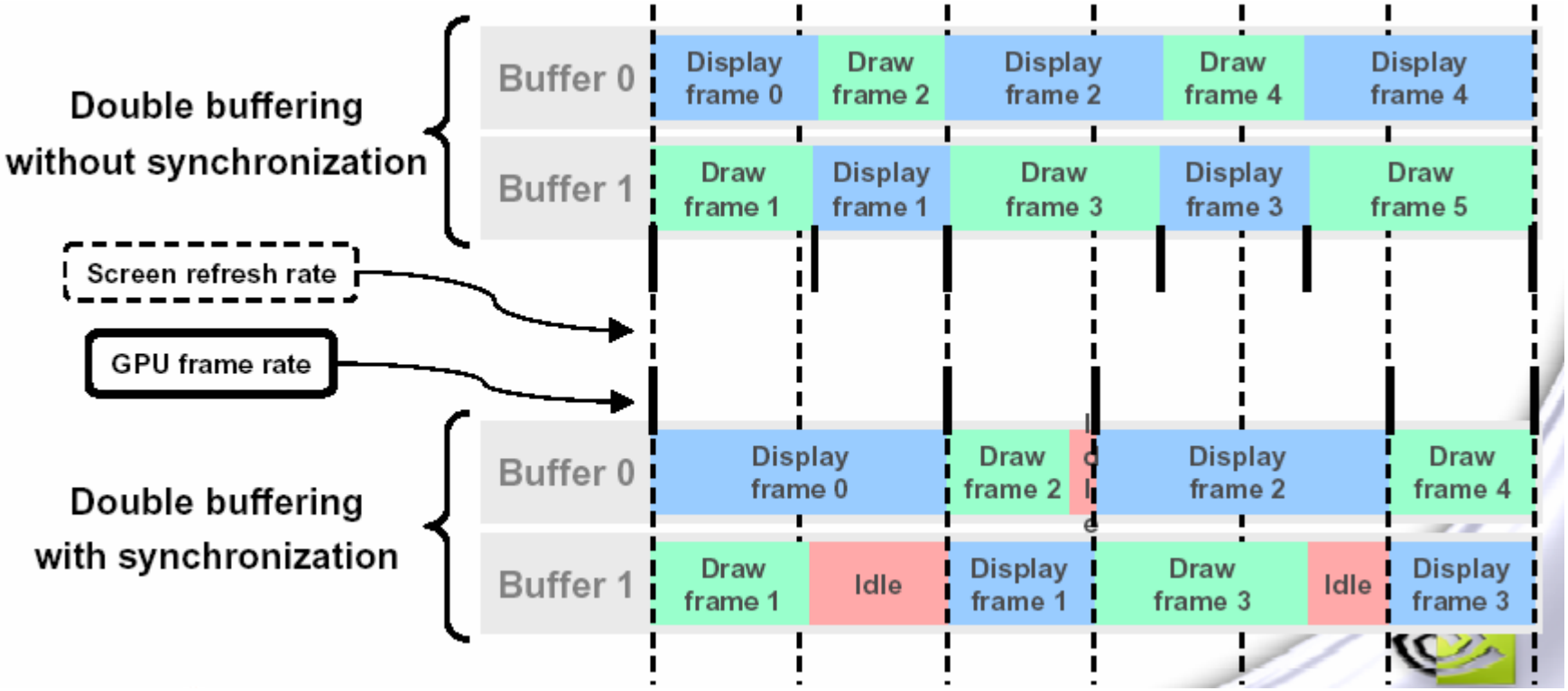
  https://open.gl/depthstencils

- brisanje spremnika i zaslona

glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

važno je brisati spremnik https://math.hws.edu/graphicsbook/demos/c3/first-cube.html

Sinkronizacija rada dvostrukog spremnika (engl. double buffer)



Kada je VSync isključen – kidanje zaslona (engl. screen tearing)

http://profs.etsmtl.ca/mmcguffin/learn/java/07-backbuffer/

https://www.zubspace.com/tools/smooth-movement-test

G-Sync (-Nvidia ili FreeSync-AMD) monitori podržavaju sinkronizaciju s GPU

2.1.2 Izlazne grafičke naprave

– podjela izlaznih grafičkih naprava

jedinice za prikaz objekata (CRT, LCD, s plazmom, pisači, crtala)

-vektorske http://groups.csail.mit.edu/graphics/classes/6.837/F01/Lecture01/Slide11.html

-rasterske http://groups.csail.mit.edu/graphics/classes/6.837/F01/Lecture01/Slide13.html

- emitirajuće (CRT, s plazmom, OLED organske diode, QLED)
- ne emitirajuće (LCD - tekući kristali)

- osvježavajuće
- s pamćenjem

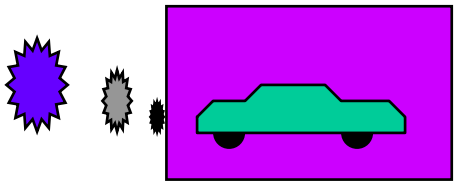
jedinice za izradu 3D objekata

- vektorski pristup (tokarilice, glodalice)
- 3D printeri - sloj po sloj

Usporedba vektorskog i rasterskog prikaza

- Vektorski (npr. SVG Scalable Vector Graphics)
  (pohrana objekata 3D mesh)
  - nekadašnja izvedba

  (ograničenost količine memorije)



  prednosti
    - točnost prikaza (ploteri)
    - jednostavna promjena mjerila

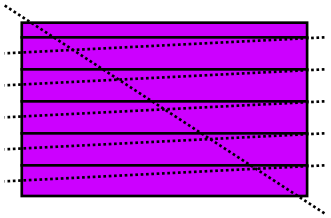
  nedostaci
    - dugačka prikazna datoteka
    - popunjavanje poligona

    ⇒ problem osvježavanja

    http://codinginparadise.org/projects/svgweb/samples/demo.html?name=tiger&svg.render.forceflash=false

    https://threejs.org/examples/webgl_loader_svg

- Rasterski
  (voxel)
  - danas uobičajeno



  prednosti
    - veličina prikazne datoteke ne utječe na frekv. osvježavanja

  nedostaci
    - potreba pretvorbe u diskretnu reprezentaciju

    => pogreška diskretizacije

    (eng. alias-sampling error
    nazubljene linije, moarè)

    http://tamats.com/

Jedinice za prikaz (različite karakteristike)

- slika se pohranjuje u slikovnoj prikaznoj memoriji
- važna je *brzina* osvježavanja zbog eksponencijalnog slabljenja intenziteta svjetla koje emitira fosfor, više kvantnih razina:
  - florescencija dio μs (snop uključen)
  - fosforescencija 10-60 μs (snop isključen)

Visoka *perzistencija* znači da svjetlu treba dugo da oslabi (manje od 10% maksimalne vrijednosti), te se tada može sporije osvježavati

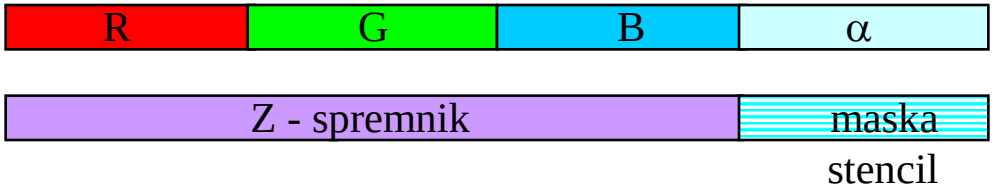
<= kontradiktorni zahtjev =>

- *dijagonala*
- *geometrijska* svojstva, omjer slike (*aspect ratio*)
- *frekvencije osvježavanja*
  - vertikalna frekvencija (broj slika u sekundi) 60-160 Hz

  (85 Hz propisano VESA standardom)
  http://www.realtimerendering.com/udacity/?load=demo/unit1-fps.js
  - horizontalna frekvencija (broj linija u sekundi) 30-100 kHz
  - frekvencija osvježavanja slikovnih elemenata (brzina paljenja i gašenja elektronskog snopa) 50-160 MHz - širina pojasa (engl. pixel rate) (http://www.colorado.edu/physics/2000/tv/moving_electrons.html)
- *razlučivost*, zrnatost, rezolucija

Pojasna propusnost prema memoriji (engl. memory bandwidth)

NPR:



2 – pristupa (piši/čitaj) = 16 bajta

1280 × 1024 × 16 bajta × 60 fps = 1,26 GB/sec.

dubinska složenost (engl. depth complexity, engl. overdraw)

1280 × 1024 × 16 bajta × 60 fps × 3 = 3,78 GB/sec.

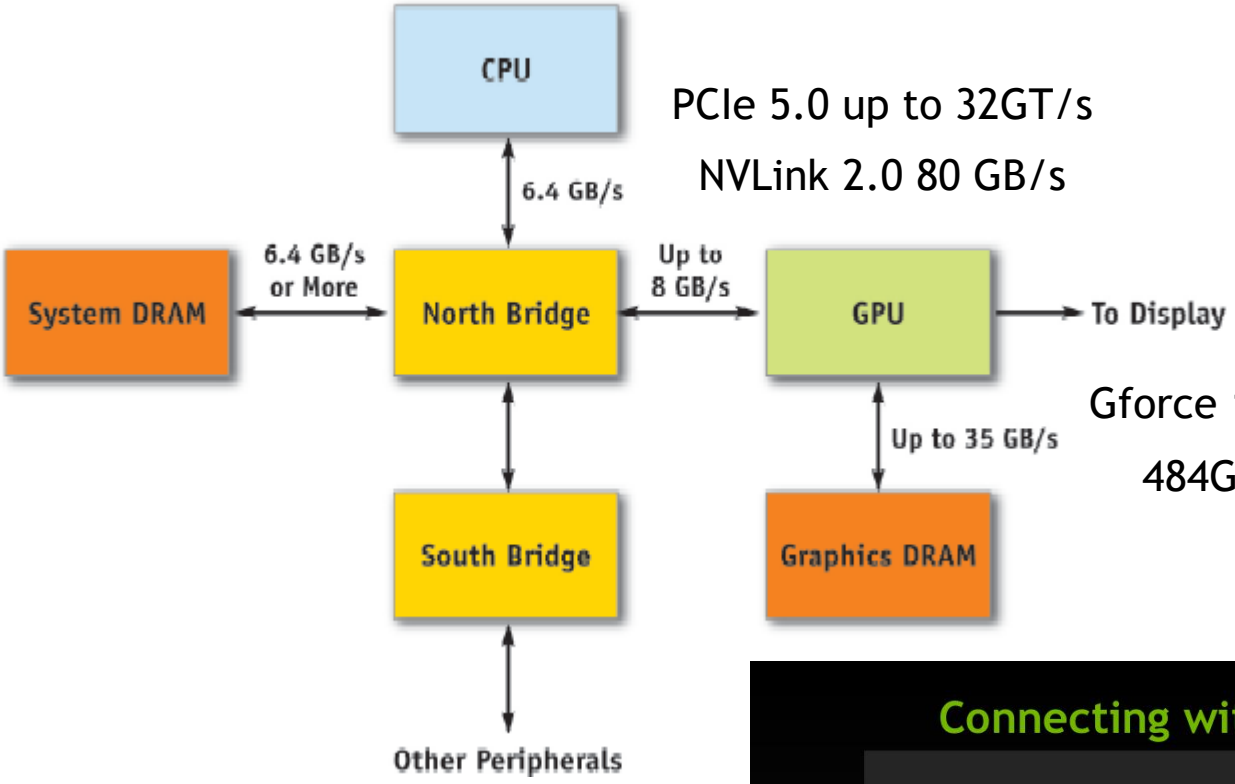
prikaz teksture – trilinearna interpolacija (8 vrhova × 4 bajta)

1280 × 1024 × (16 + 32) bajta × 60 fps × 3 = 11,32 GB/sec.

antialias × 4 (engl. FSAA Full Screen Antialiasing)

1280 × 1024 × (16 + 32) bajta × 60 fps × 3 × 4 = 45,3 GB/sec.

Primjer: Pojasna propusnost prema memoriji (engl. memory bandwidth)

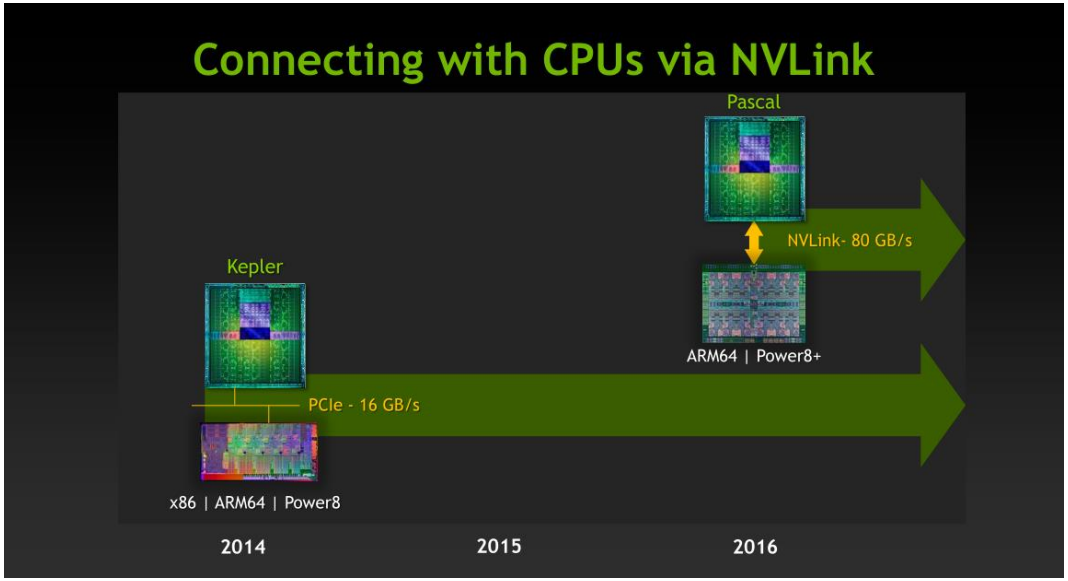


PCIe 5.0 up to 32GT/s

NVLink 2.0 80 GB/s

Gforce 1080Ti

484GB/s

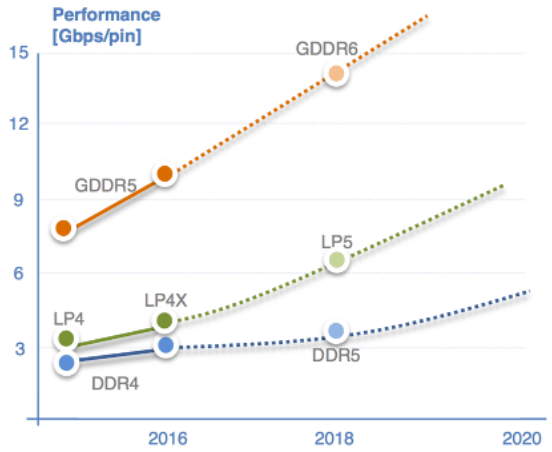


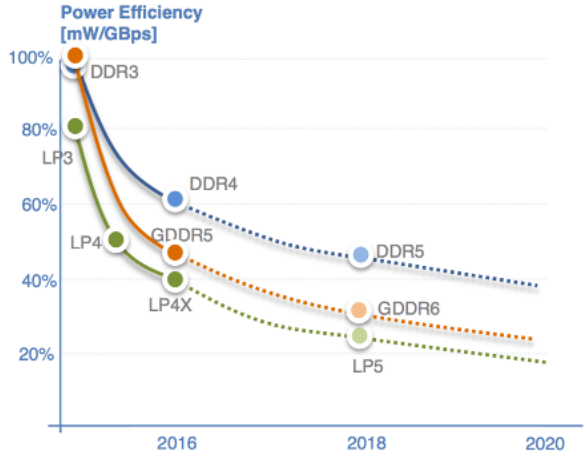
- raspoložive *memorije* (https://ourworldindata.org/grapher/historical-cost-of-computer-memory-and-storage?time=1956..1990)

  - SDRAM, (interno paralelna organiz.)

  - SGRAM (synchronous graphics RAM, ima dodatne grafičke mogućnosti, može biti i dvopristupni (engl. dual port))

  - DRAM (engl. dynamic)





VRAM (Video RAM):

- DDR
(engl. Double-Data-Rate)
- GDDR (Grafički DDR)
- LP (engl. low power)
- HBM
(engl. High-Bandwidth Memory)

Trend je ostvariti ***In-Memory*** procesiranje –tj. obradu podataka ostvariti u memoriji

- povećati paralelizam i smanjiti tok podataka na sabirnici (APU = CPU + GPU)

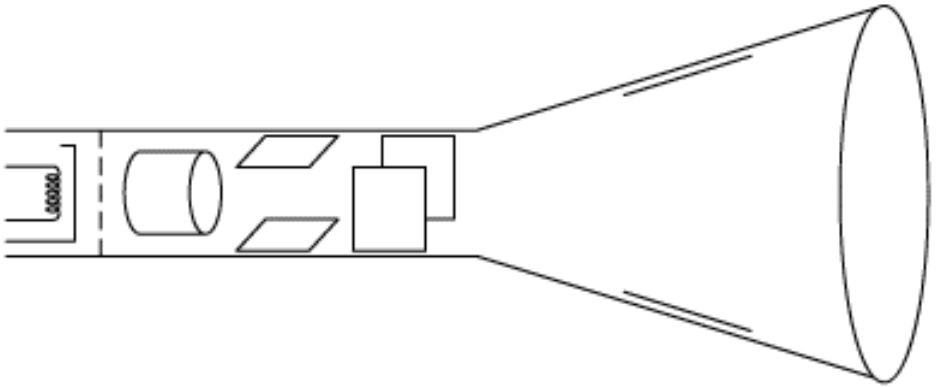
JEDINICE ZA PRIKAZ

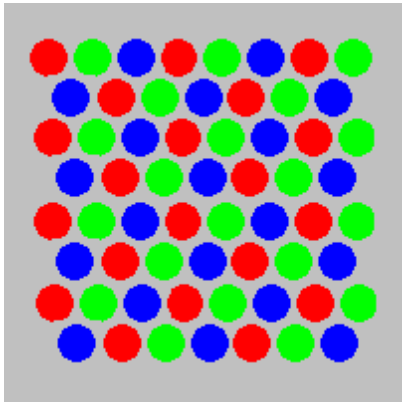
CRT (princip rada) – snop elektrona pogađa *fosfor* naparen na staklo - emitira se *svjetlo* određene valne duljine r, g, b - miješanje valnih duljina => *oko* čovjeka

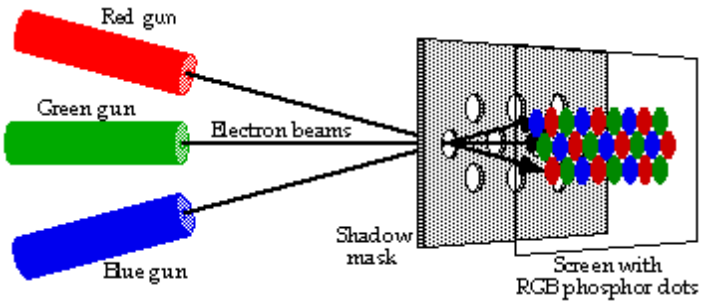
-utjecaj ambijentnog svjetla na svjetlinu i kontrast

https://phet.colorado.edu/sims/html/color-vision/latest/color-vision_en.html

https://michaelbach.de/ot/lum-adelsonCheckShadow/index.html

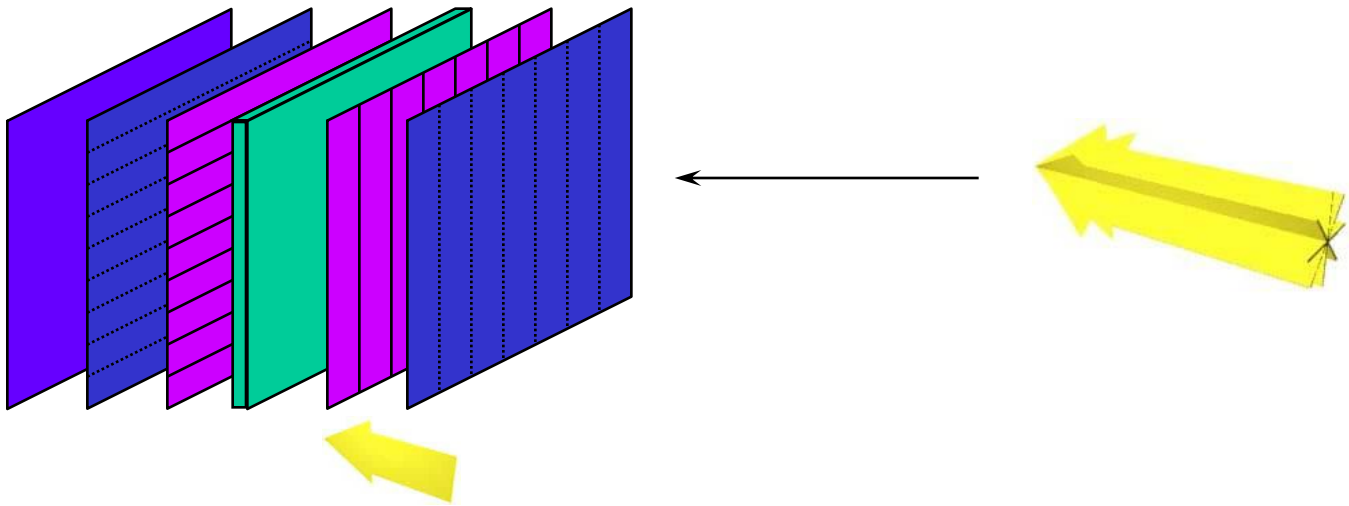






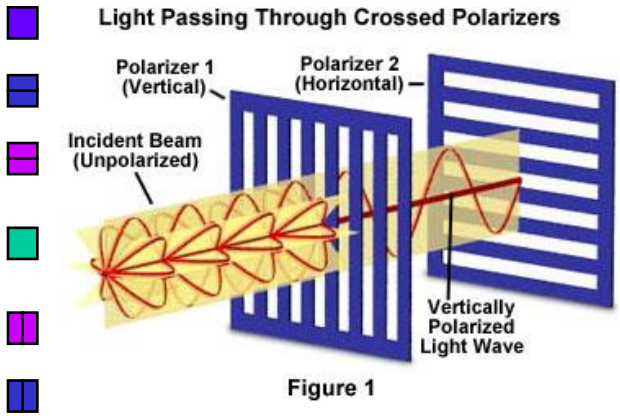
LCD prikazna jedinica s tekućim kristalima

- https://www.edumedia-sciences.com/en/media/37-polarizer



- šest slojeva :

reflektirajući sloj
horizontalna polarizacija
horizontalne žičice
sloj tekućih kristala
vertikalne žičice
vertikalna polarizacija



princip rada

- materijal tekućih kristala je načinjen od dugačkih molekula

  - kada je kristal u *električnom polju* nema polarizirajuća svojstva na svjetlo koje dolazi, pa svjetlo ostaje vertikalno polarizirano i *ne prolazi* kroz horizontalnu polarizaciju

  - kada je kristal *nije* u električnom polju *zakreće* ravninu polarizacije za 90º iz vertikalne u horizontalnu

- https://ophysics.com/l3.html
- TFT - (eng. thin film transistor) na svakom (x, y) ima tranzistor, služe kao aktivna memorija dok se stanje ne promijeni

prednosti - lagani, mala potrošnja, mali po z-osi,

nedostaci - nisu izvor svjetlosti no može se koristiti stražnje osvjetljenje za projekcije, spora promjena slike, kut gledanja je ograničen, osjetljivi na pritisak i visoku temperaturu

upotreba
- prijenosna računala
- projektori
- HMD

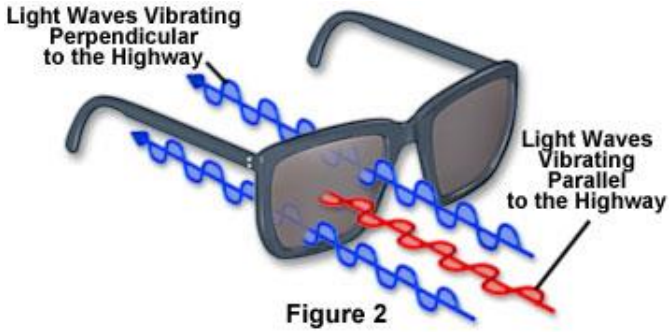
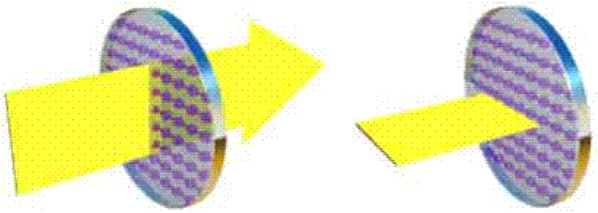
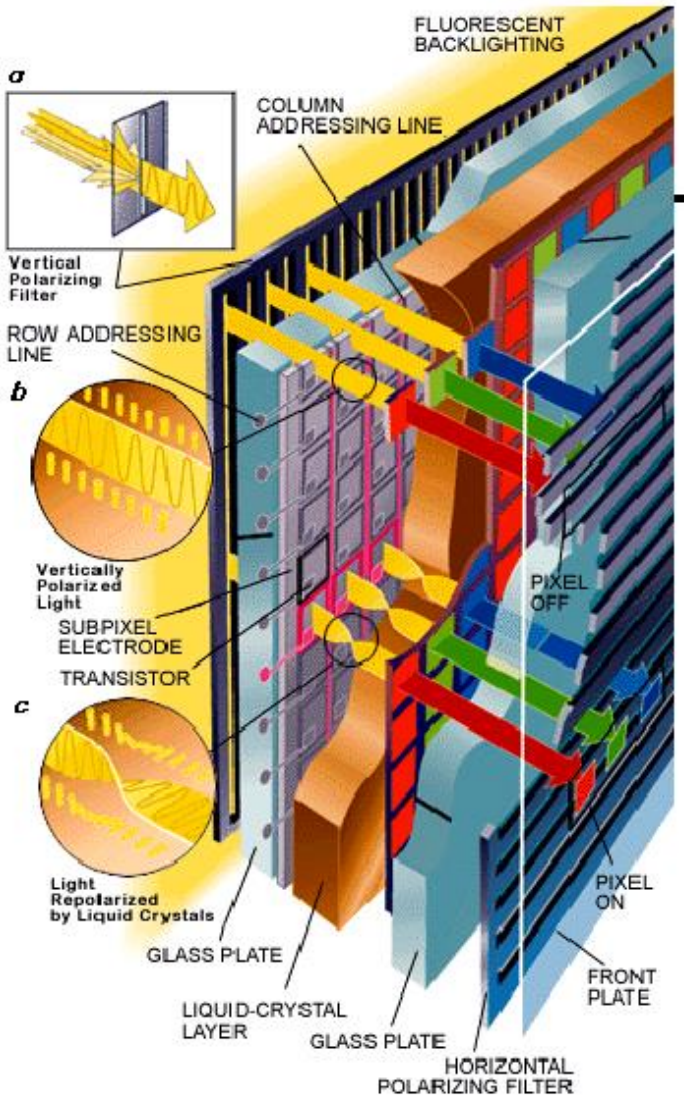


Figure 2

LCDs





TN - Twisted Nematic

- okomito polje
- jeftin, brz,
  mali kut, 6-bita boja (dithering prostorni i vremenski za 8-bita)

IPS - In Plane Switching

- horizontalno polje
- velik kut (TV), 8-bita boja,
- sporiji, skuplji, jače pozadinsko svjetlo

PVA - Vertical Alignment

- kristali okomito na zaslon kao TN a međusobno paralelni -IPS
- brzi, veliki kut, boje, kontrast
- skuplji (Samsung)

http://micro.magnet.fsu.edu/primer/java/scienceopticsu/polarizedlight/filters/index.html

https://www.olympus-lifescience.com/en/microscope-resource/primer/java/polarizedlight/3dpolarized/

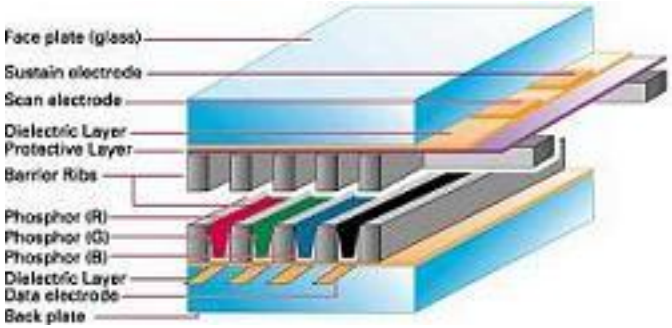
Prikazna jedinica s **plazmom**

- zapravo mali CRT za svaki slikovni element
- LCD prikazne jedinice nisu izvor svjetlosti
- na mjestu ukrštanja elektroda je adresirano mjesto zatim dolazi do ionizacije xenon/neon (xenon/neon XeNe) plina, to izaziva ultravioletno zračenje koje aktivira fosfor - svjetlo (nije pasivni uređaj)
- jedinice s plazmom mogu imati veličinu ~ 40'', 61'' (-100'')
- nedostatci – veliki slikovni elementi (1 mm, CRT 0,2 mm),

  vakuum u malim fluorescentnim cijevima – deblje staklo

  velika potrošnja (40'' ~ 300W) uz slabu svjetlinu (~ 1/3 CRT),

https://phet.colorado.edu/en/simulations/molecules-and-light Vis NO2, UV O3



OLED, QLED prikazna jedinica

– OLED (engl. Organic Light-Emitting Diode Arrays) http://transparent-

+ tanki, fleksibilni, dobra svjetlina – izvori svjetla, dobar kontrast,

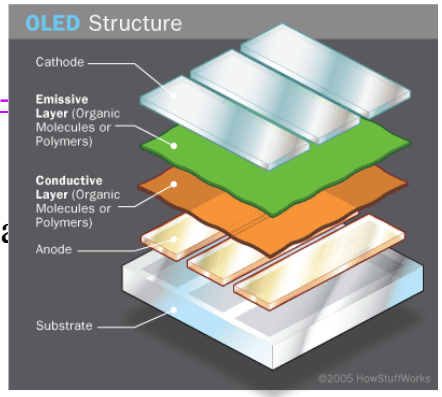
malo troše, veliki kut pogleda, kvaliteta crne boje, malo vrijeme odziva

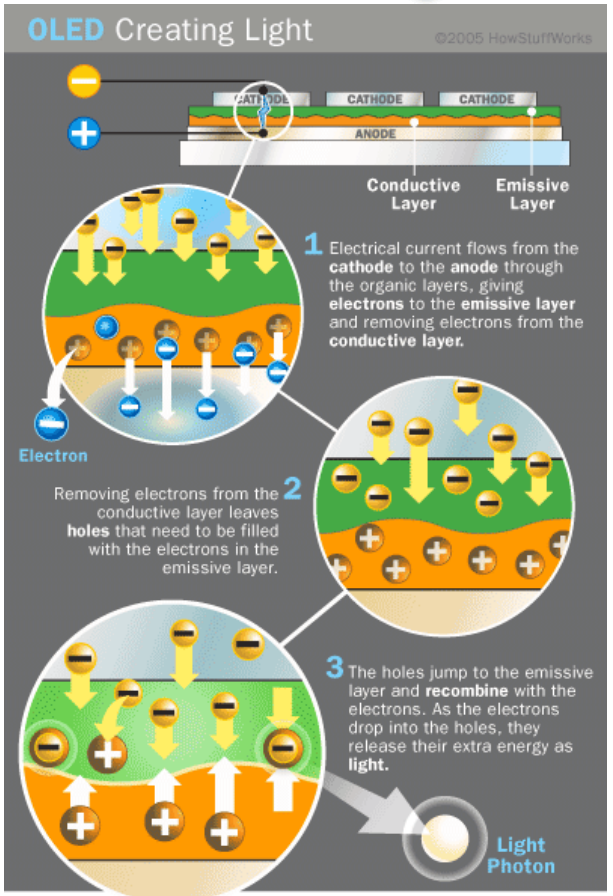
- vijek trajanja, osjetljivi na vodu

AM OLED – s aktivnom matricom (za dodirne ploče)

- postaju dominantni na tržištu

- QLED (engl. Quantum dot display)
  - Q – kvantna točka (nanočestice ovisno o veličini kreiraju boju, velike crvenu, male plavu - zasićenije boje, gušće)
  - bolji prikaz raspona boja
  - manje podložni *burn-in* učinku





OLED Display Screen (from Universal Display Corp)

3D slike na 2D uređaju:

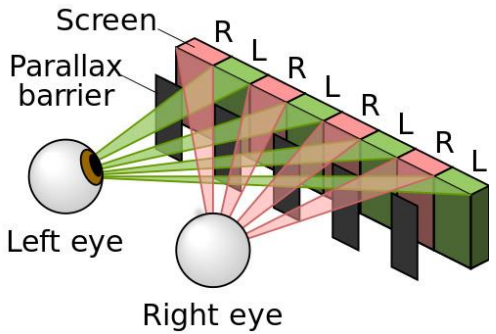
anaglyph (crvena/zelena slike) http://threejs.org/examples/#webgl_effects_anaglyph

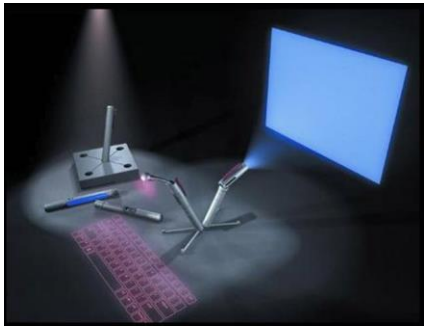
parallaxbarrier http://threejs.org/examples/#webgl_effects_parallaxbarrier

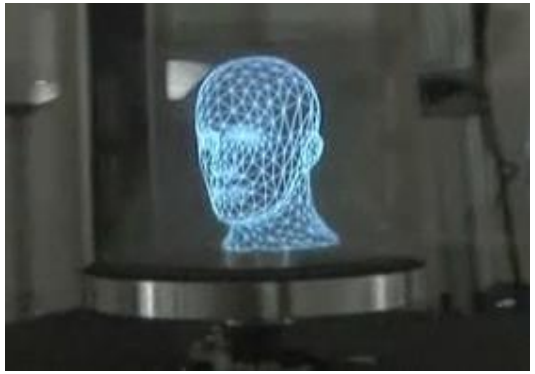
Druge tehnologije:

HoloLens (CPU, GPU, HPU - Holographic proc. unit

24 Tensilica DSP procesira signale s akcelerometra …)



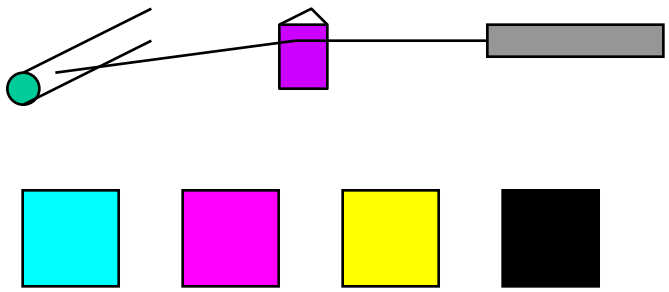




Jedinice za izradu 3D objekata

po uzoru na pisače (printeri)

- matrični
- laserski
- ink-jet
- termo
  - CMYK (više prolaza)



- izrada objekata sloj po sloj
  - 3D pisači (ZPrinter)
  - uređaji za stereolitografiju

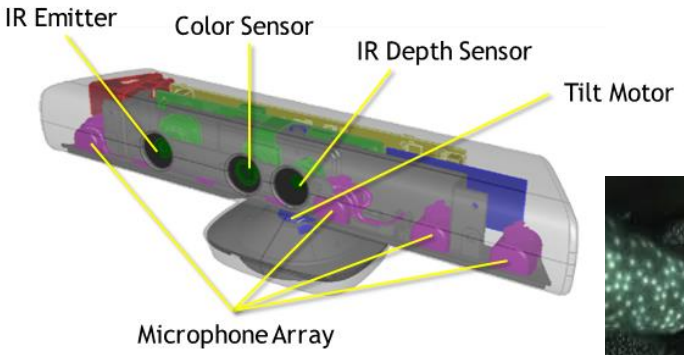
(važno za brzu izradu prototipa)

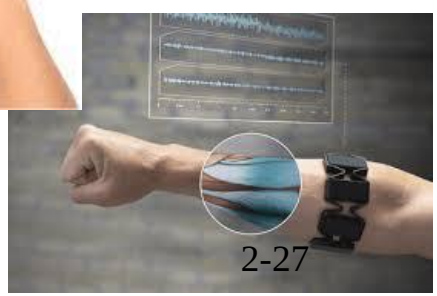
- tokarilice, glodalice

2.2.3 Ulazne grafičke naprave

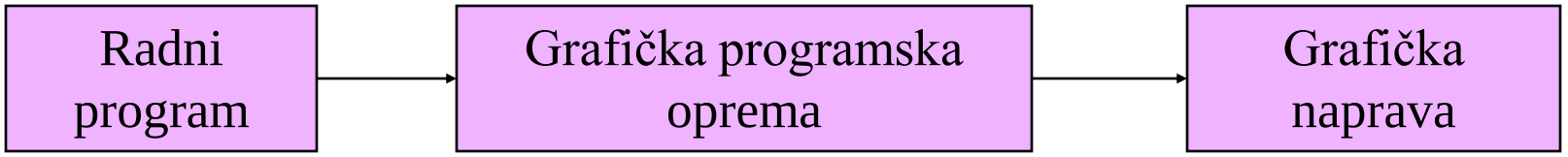
- tablica (engl. tablet) s pisaljkom, na dodir osjetljiva ploča
- sustavi za 3D uzorkovanje
  - laserskim snopom - mogućnost uzorkovanja boje i temperature

    Npr: http://www.cc.gatech.edu/projects/large_models/
  - Kinect – infracrveni dio spektra
  - LeapMotion, Myo (geste) http://edankwan.com
  - LiDAR (3D Laser Mapping)
  - CT, PET (računalna tomografija)
  - MR (magnetska rezonancija)





2.2 PROGRAMSKA GRAFIČKA OPREMA



- Knjižnica grafičkih rutina (grafičke rutine koje se pozivaju iz nekog višeg programskog jezika s atributima C, C++). Teži se da ova knjižnica bude načinjena prema specifikaciji tj. prema nekom standardu.

  "+" neovisnost radnog programa o sklopovskoj opremi

  "-" obično se ne može ostvariti potpuna iskorištenost sklopovske opreme
- Standardima je propisano
  - API (*Application Programming Interface*) definiran specifikacijom
    - OpenGL (*cross-language*, *cross-platform*),
    - DirectX Direct3D
  - zapisi
    - slika TIF, GIF, BMP, JPEG, DDS, PS (rasterski, vektorski)
    - niza slika GIF, video AVI, MOV, WMV, MPG, MP4, SWF, RM
    - scene, objekti DXF, MAX, 3DS, WRL-vrml, PLY, OBJ, FBX COLLADA (.dae), glTF, USD

- Grafičke jezgre načinjene u okviru standarda
  - 3D CORE (Core Graphics System)
    - 1979. ACM SIGGRAPH (Association for Computing Machinery Special Interest Group on Graphics)
  - GKS (Graphics Kernel System)
    - ISO 88, 94, 97, 98, 99 (International Standards Organization)
    - ANSI 85 (American National Standards Institute)
  - PHIGS (Programmer's Hierarchical Interactive Graphics System),
  - VRML (Virtual Reality Modelling Language)
    - ISO 97, 98, 99

- Osim službenih standarda postoje "de facto" ili industrijski standardi
  - 93' GL, OpenGL — SGI
  - 95' Direct 3D — Microsoft
  - RenderMan — Pixar
  - PostScript — Adobe
  - OpenFlight
  - WebGL

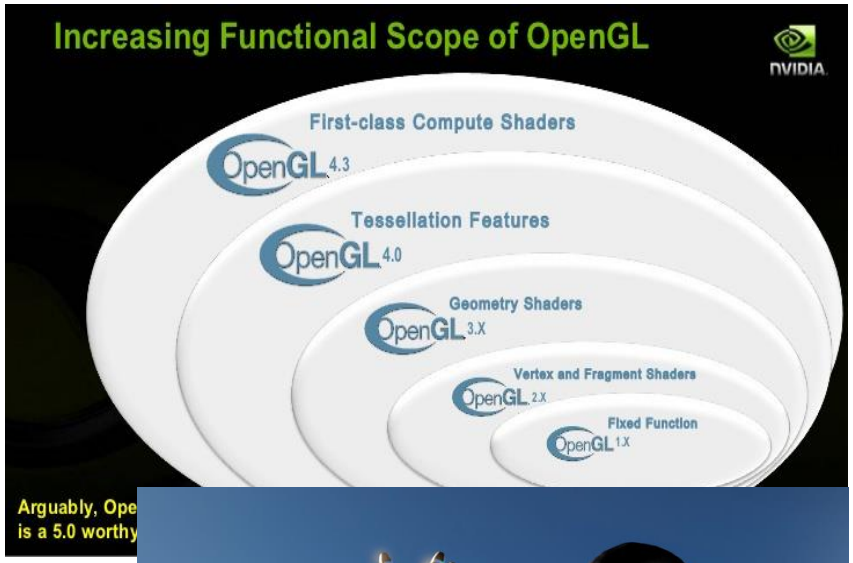
  Khronos, Open standard APIs - 2000

Komercijalno su ovi standardi značajniji od službenih jer se jednostavnije mogu mijenjati.

OpenGL - SIGGRAPH

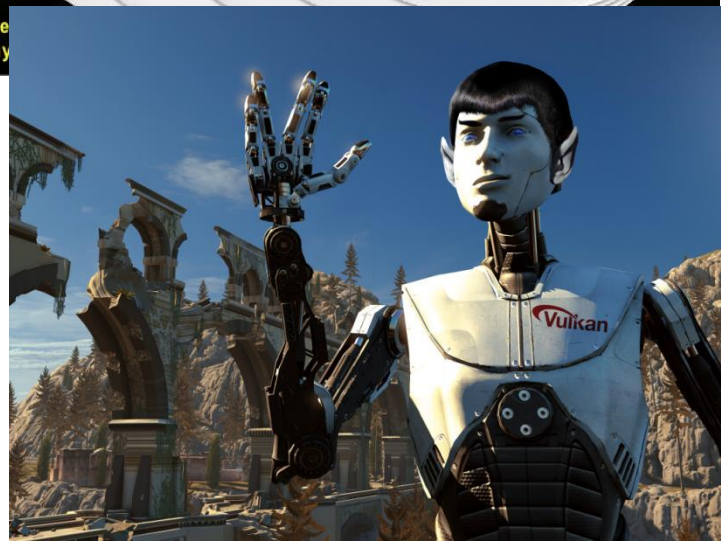
- 2001. OpenML – integracija i sinkronizacija 3D grafike s video i audio zapisima (Media - rich programming, Khronos group)
- 2003. OpenES – podrška za ugrađene sustave (embedded 3D graphics)
- 2004. OpenGL 2.0 (GLSL)
- 2008. OpenGL 3.0 (*geometry shad.*)
- 2010. OpenGL 4.0
- 2015. Vulkan

(GLEW – isčitamo što nam je dostupno od ekstenzija)



DirectX – Microsoft

- 2000. Direct3D 8 (*vertex and pixel shaders)*
- 2002. Direct3D 9 (HLSL)
- 2007. Direct3D 10 (*geometry shaders*)
- 2009. Direct3D 11
- 2014. Direct3D 12



Jezici i tehnologije za paralelno programiranje

OpenCL – (Open Computing Language)
- za razne platforme

CUDA (Nvidia)

- http://developer.nvidia.com/object/cuda.html

Jezici za sjenčanje (engl. shading languages)

- programiranje grafičkog sklopovlja korištenjem jezika više razine (kako ne bi morali programirati u asembleru za karticu)

HLSL (engl. High-Level Shading Languages) – Direct3D, Microsoft, ’02.

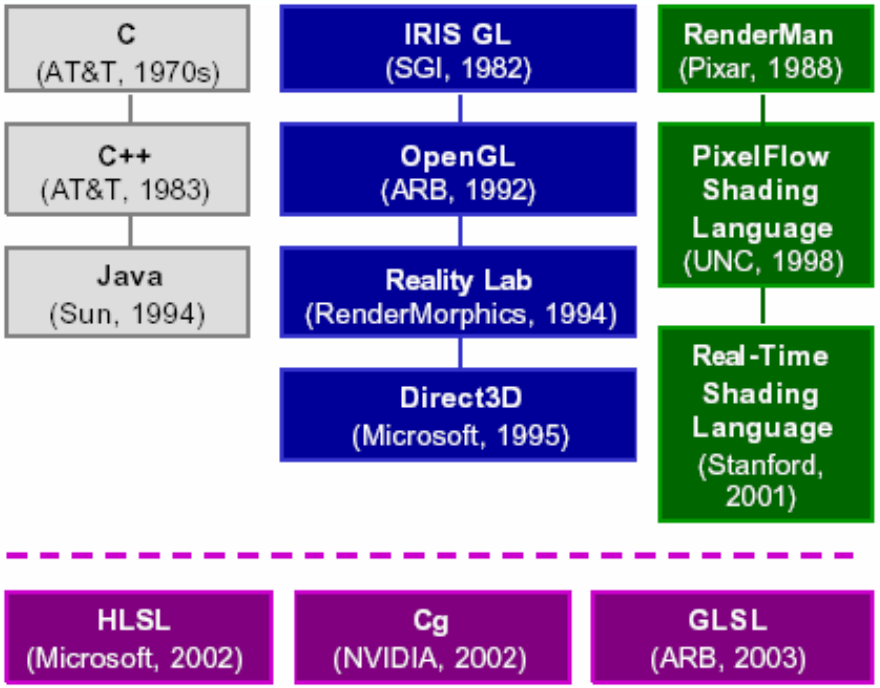
CG (engl. C for graphics) – Opengl, Direct3D, NVidia, ’02.

GLSL (engl. The OpenGL Shading Language) – open standard, ARB, ’03.

- **Alati** za izradu programa u SL FX-composer (NVidia – HLSL)

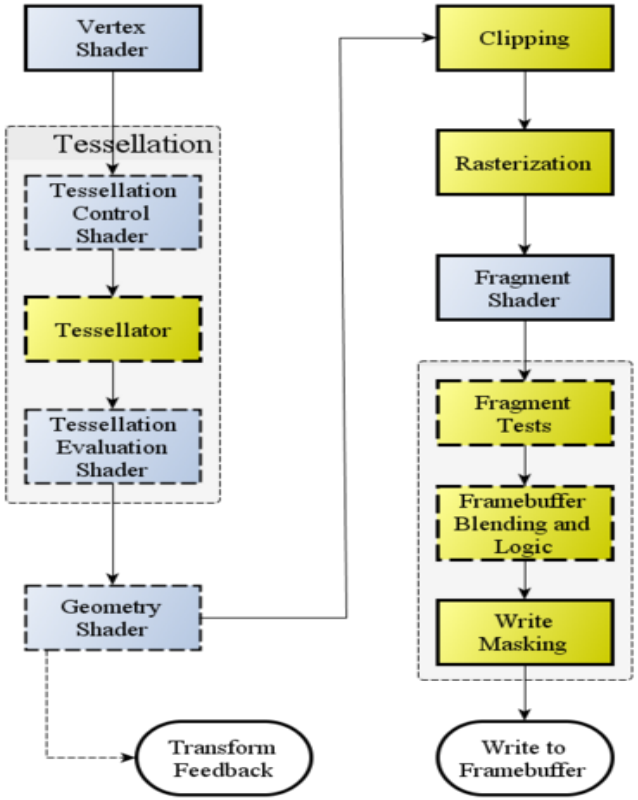
http://developer.nvidia.com/object/fx_composer_home.html

RenderMonkey (AMD, ATI)



Jezici za sjenčanje
(engl. Shading Languages):

Compute Shaders



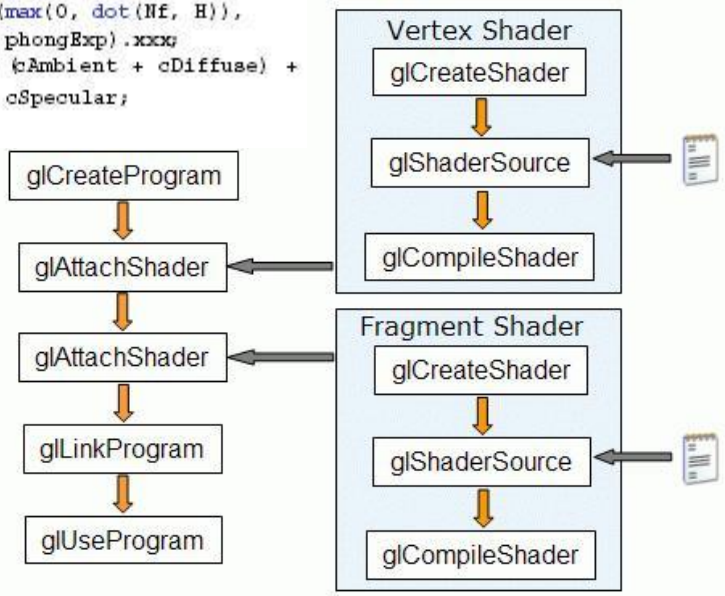
Assembly

```
...
DP3 R0, c[11].xyzx, c[11].xyzx;
RSQ R0, R0.x;
MUL R0, R0.x, c[11].xyzx;
MOV R1, c[3];
MUL R1, R1.x, c[0].xyzx;
DP3 R2, R1.xyzx, R1.xyzx;
RSQ R2, R2.x;
MUL R1, R2.x, R1.xyzx;
ADD R2, R0.xyzx, R1.xyzx;
DP3 R3, R2.xyzx, R2.xyzx;
RSQ R3, R3.x;
MUL R2, R3.x, R2.xyzx;
DP3 R2, R1.xyzx, R2.xyzx;
MAX R2, c[3].z, R2.x;
MOV R2.z, c[3].y;
MOV R2.w, c[3].y;
LIT R2, R2;
...
```

https://webglreport.com/

High-Level Language

```
...
float3 cSpecular = pow(max(0, dot(Nf, H)),
                       phongExp).xxx;
float3 cPlastic = Cd * (cAmbient + cDiffuse) +
                  Cs * cSpecular;
...
```



Vertex Shader

```
void main() {
  gl_Position = gl_Vertex;
}
```

Fragment Shader

```
void main() {
  gl_FragColor = vec4(1, 0, 0, 1);
}
```

OpenGL (engl. Open Graphics Library)

http://www.opengl.org/

state.pdf

- IrisGL - SGI temelj za OpenGL, 1992. industrijski standard
- stroj stanja (engl. state machine) koji kontrolira skup specifičnih operacija crtanja 2D/3D (definira kontekst za prikaz)
- OpenGL se temelji na spremniku ***FrameBuffer*** no u svom konceptu ne podržava grafičke ulazno izlazne naprave kao što su miš ili tipkovnica
- programsko **sučelje prema grafičkom sklopovlju**, **neovisan o platformi** tj. o OS-u i grafičkom sučelju prozora (engl. **window system**)

Dodatne biblioteke

- u pravilu otvorenog koda, podržavaju razne platforme i operacijske sustave (MS Windows, Linux, FreeBSD), programske jezike (Ada, C++, C#, Common Lisp, Go, Haskell, Java, Python, Ruby)

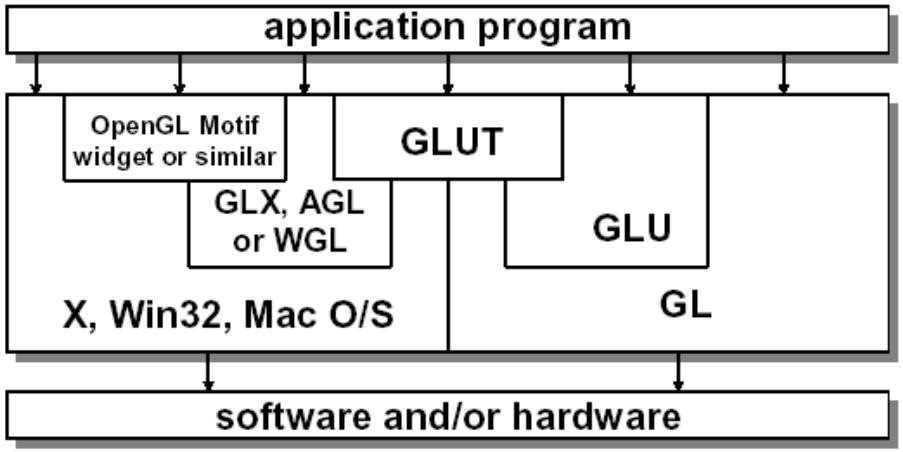
GLU Utility Library (pomaže u modeliranju i nekim operacijama s prozorima)

- objekti (kugla, cilindar, čajnik)
- dijeljenje poligona, NURBS
- od verzije OpenGL 3.x+ (ugrađeno u sam standard)

- FREEGLUT → GLUT (OpenGL Utility Toolkit, truly cross-platform)
  - neslužbeni dio OpenGL, pisanje prenosivih aplikacija koje rade u okruženju prozora
  - pojednostavnjuje stvaranje prozora, rukovanje događajima (engl. events)

```
#include <GL/gl.h>
#include <GL/glu.h>
#include <GL/glut.h>        // uključuje gl.h i glu.h pa ih nije potrebno navoditi
```

- GLFW – slično kao GLUT, API za izradu prozora, OpenGL konteksta, U/I
- GLEW (OpenGL Extension Wrangler Library) http://glew.sourceforge.net/
  - OpenGL je zapravo specifikacija – Khronos konzorcij tvrtki (ne i implementacija)
  - GLEW olakšava korištenje OpenGL ekstenzija - odnosno učitava pokazivače na dostupne funkcije, mogu se isčitati tijekom izvođenja programa
  - Glad – daje OpenGL funkcionalnosti dostupne programeru.



Programčić koji crta kvadrat:

```
#include <GL/glut.h>

void crtaj() {
    glClear(GL_COLOR_BUFFER_BIT);
    glLoadIdentity();
    glOrtho(0, 4, 0, 4, -1, 1);
    glBegin(GL_POLYGON);
        glVertex2i(1, 1);
        glVertex2i(3, 1);
        glColor3f(0.5, 0, 0.5);
        glVertex2i(3, 3);
        glVertex2i(1, 3);
    glEnd();
    glFlush();
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_RGBA); //GL_DEPTH_BUFFER
    glutCreateWindow ("kvadrat");
    glutDisplayFunc (crtaj);
    glutMainLoop();
}
```

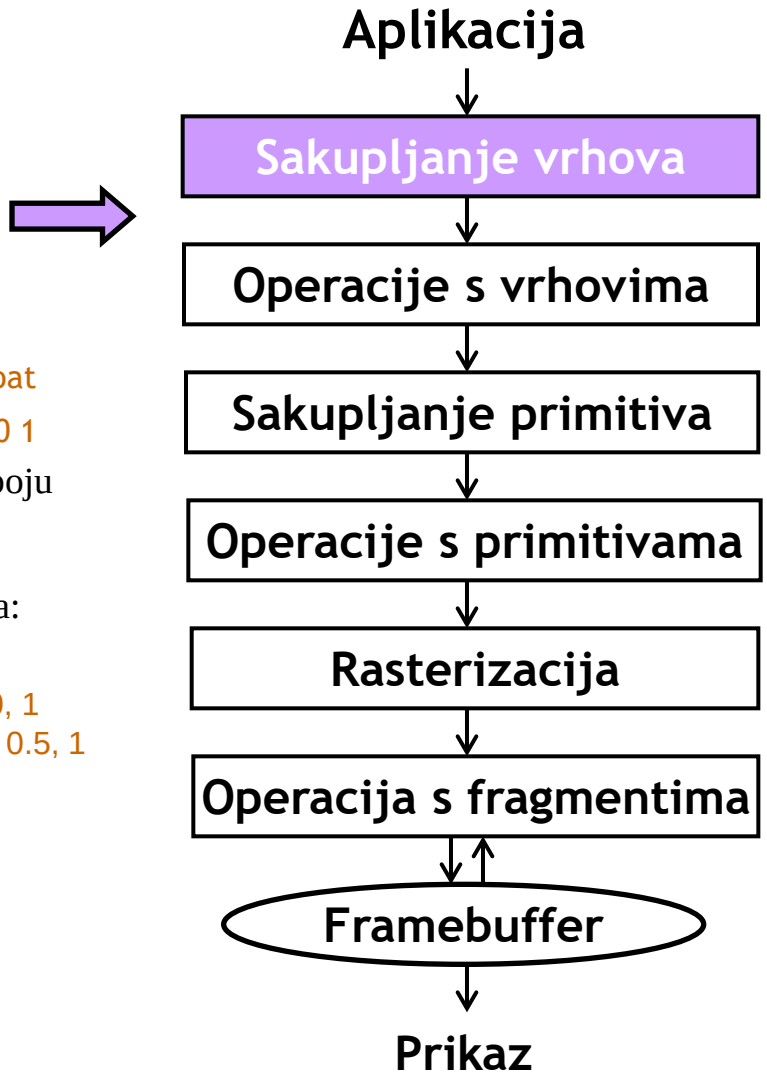
konverzija u internu reprezentaciju npr:

glVertex2i(3, 3); u float z, w - inicijalizira na 0 1

postavlja stanje npr. boju

interna reprezentacija:

```
struct {
  float x,y,z,w; // 3, 3, 0, 1
  float r,g,b,a; // 0.5, 0, 0.5, 1
} vertex;
```



Operacije s vrhovima (procesor vrhova):

- transformiranje vrhova
  (množenje transformacijskom matricom)
- proračun osvjetljenja u vrhovima
- proračun koordinata teksture
- …

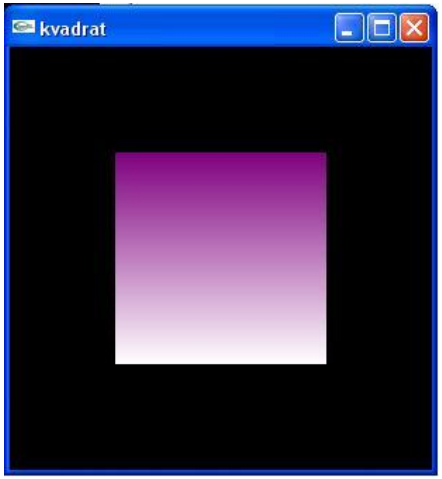
skalira koordinate obzirom na veličinu prozora

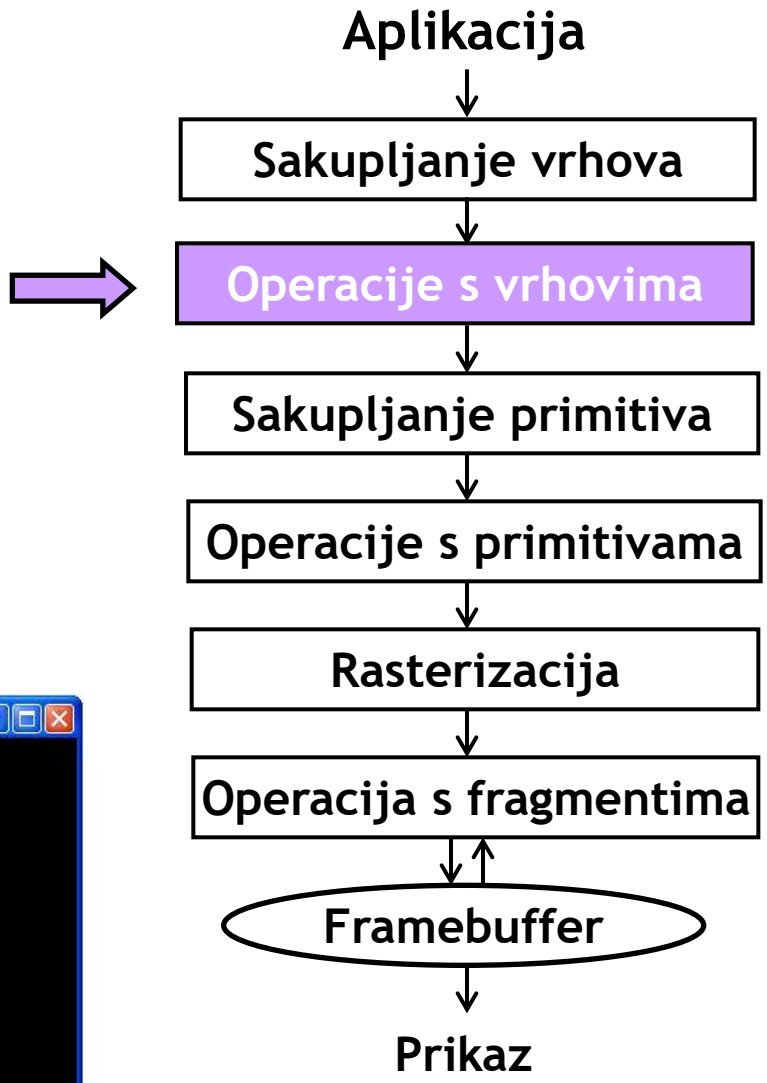
U našem slučaju:

- skaliranje koordinata na veličinu prozora

```
glLoadIdentity();
glOrtho(0, 4, 0, 4, -1, 1);
```

- veličine koje nismo odredili
  npr. dimenzije prozora su predefinirane
  (default npr. 300×300)





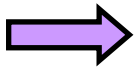
Sakupljanje primitiva:

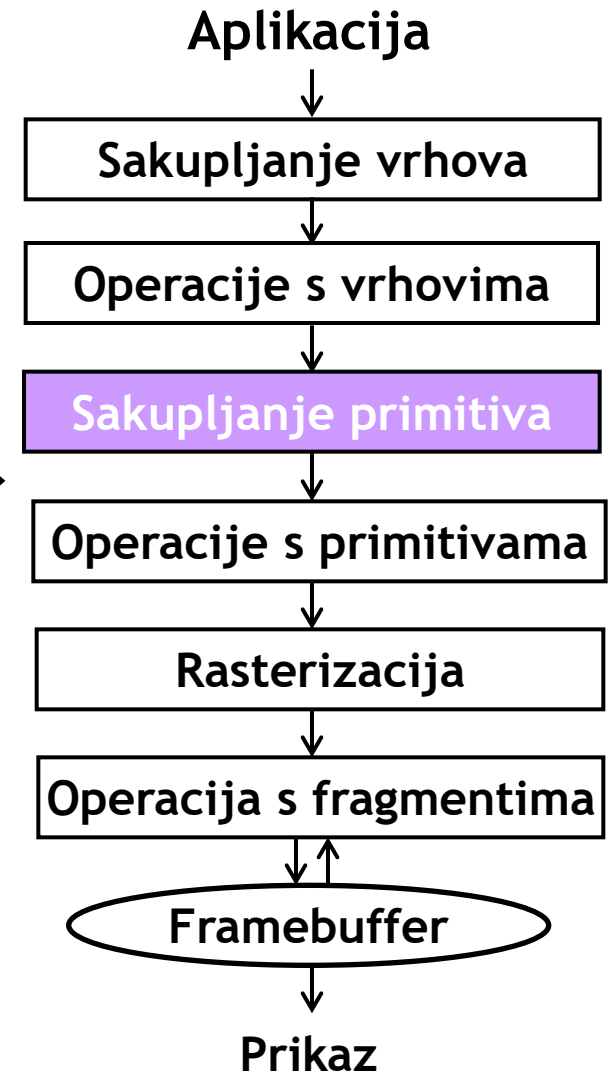
- povezivanje vrhova primitivama
- primitive su točke, linije, trokuti, niz trokuta, …

```
glBegin(GL_POLYGON);
    glColor3f(1, 1, 1);
    glVertex2i(1, 1);
    glColor3f(1, 0, 0);
    glVertex2i(3, 1);
    glColor3f(0, 1, 0);
    glVertex2i(3, 3);
    glColor3f(0, 0, 1);
    glVertex2i(1, 3);
glEnd();
```

rastavljanje poligona na niz trokuta

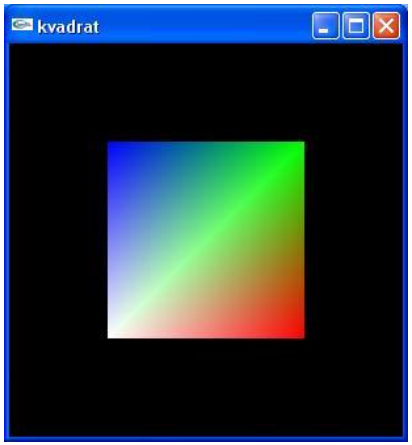
```
struct {
  vertex v0,v1,v2
} triangle;
```

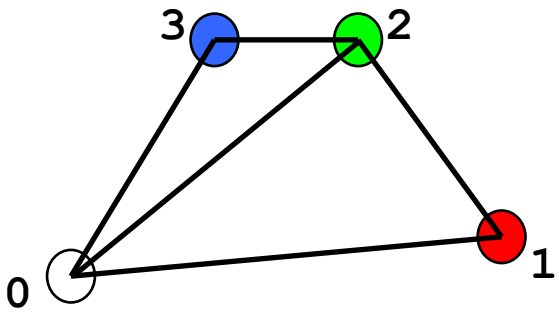




U našem slučaju:

- poligon se rastavlja na niz trokuta





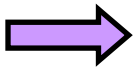
Operacije s primitivama:

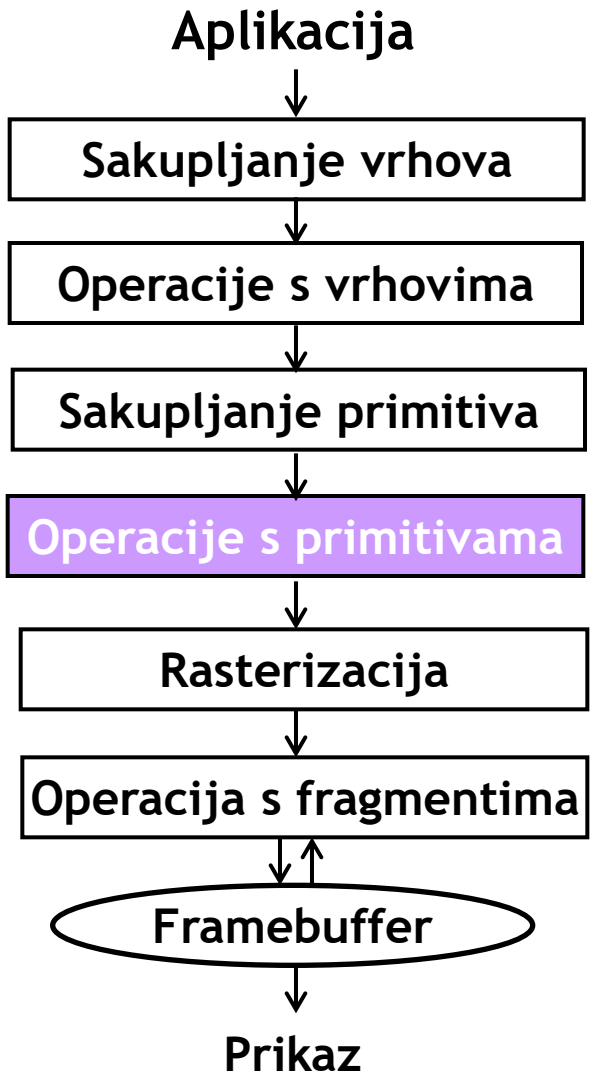
- odsijecanje obzirom na prozor

odnosno prema piramidi pogleda (frustum surfaces)

- uklanjanje stražnjih poligona (Culling)

odsijecanje (clipping)





U našem slučaju:

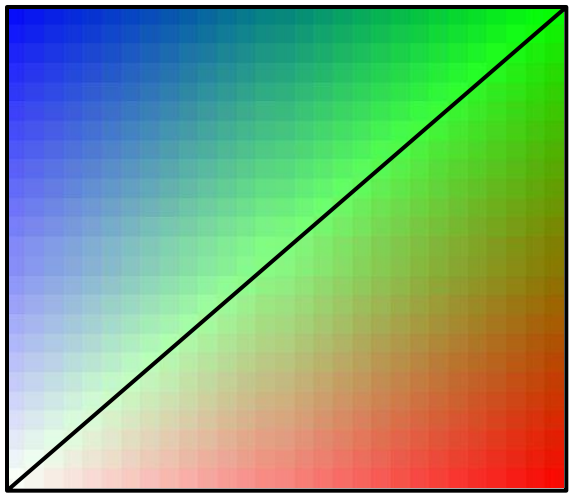
- ništa (samo se provjerava)

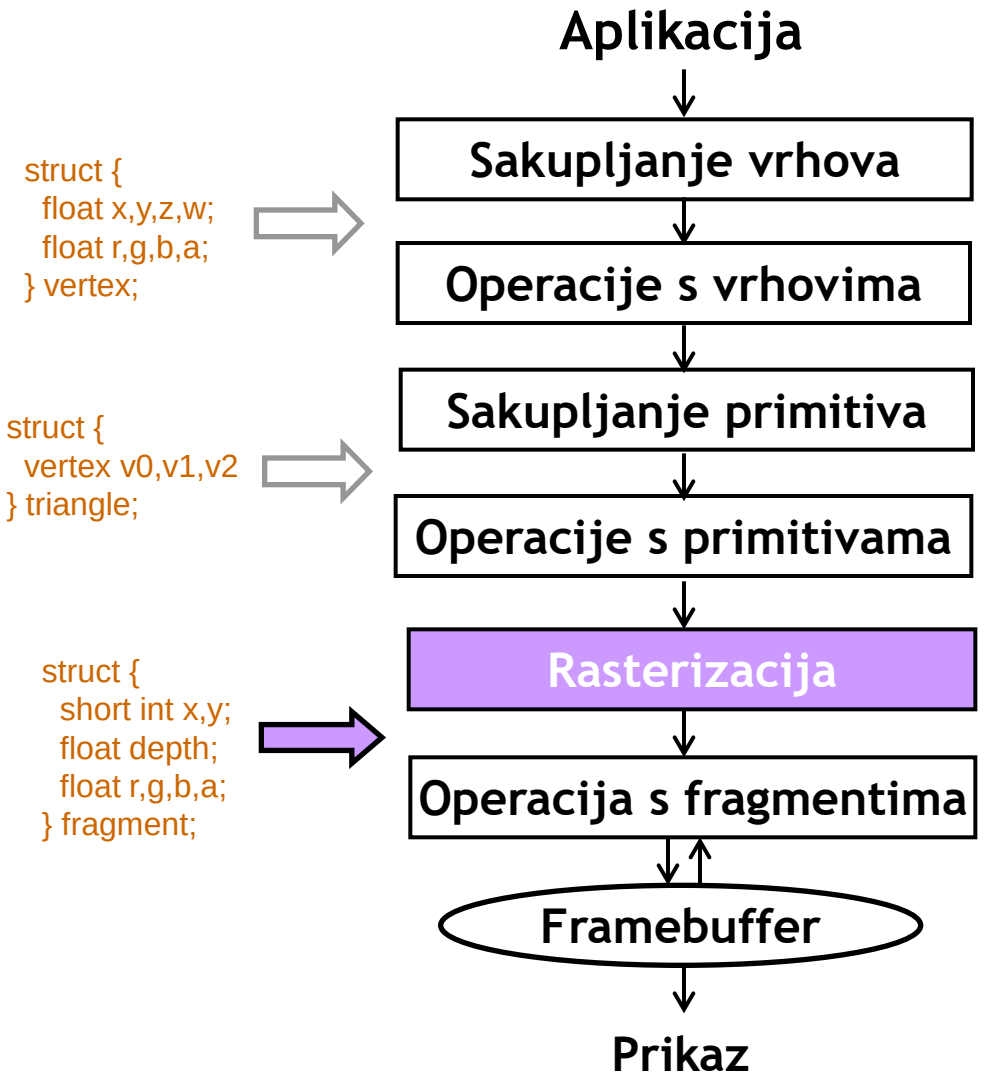
- ako bi neki vrh bio van prozora tada se odsijeca

Rasterizacija:

- određuje se koji slikovni elementi čine primitivu
- stvaranje fragmenta za svaki sl. el.
- pridruživanje atributa (npr. boja) svakom fragmentu

U našem slučaju:





Operacije s fragmentima (fragmet shader):

- preslikavanje teksture, magla,
- proračun osvjetljenja po fragmentu

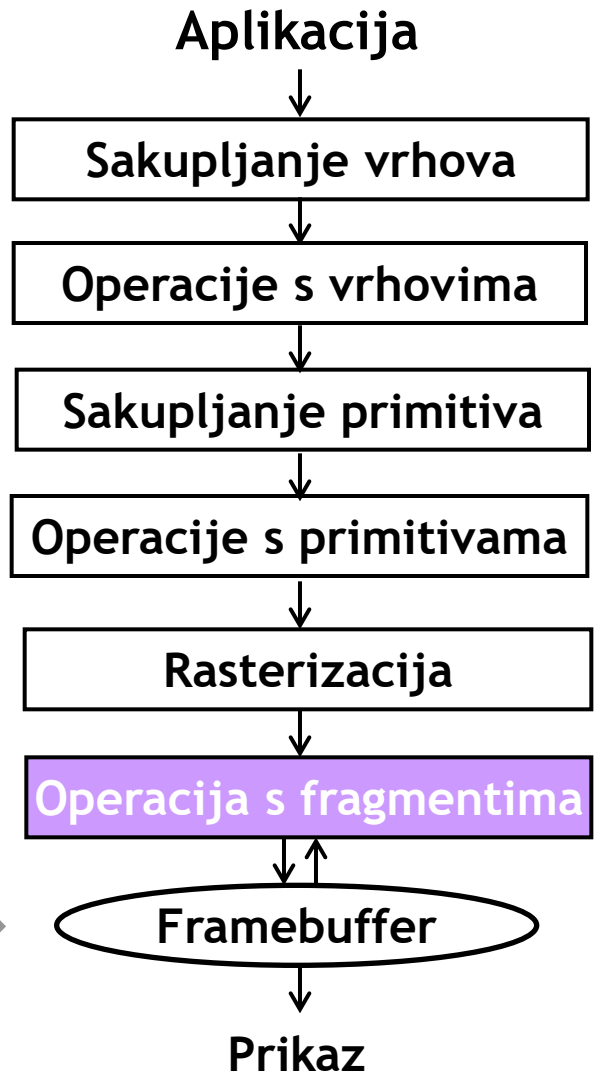
(različiti **l**, **n** vektori prema izvoru i vektor normale za svaki slikovni element)

Veza sa slikovnom memorijom (FrameBuffer)

- konačna slika se gradi i sastavlja ovisno o Z-spremniku, miješanju boja (color blending), …

U našem slučaju:

- ništa



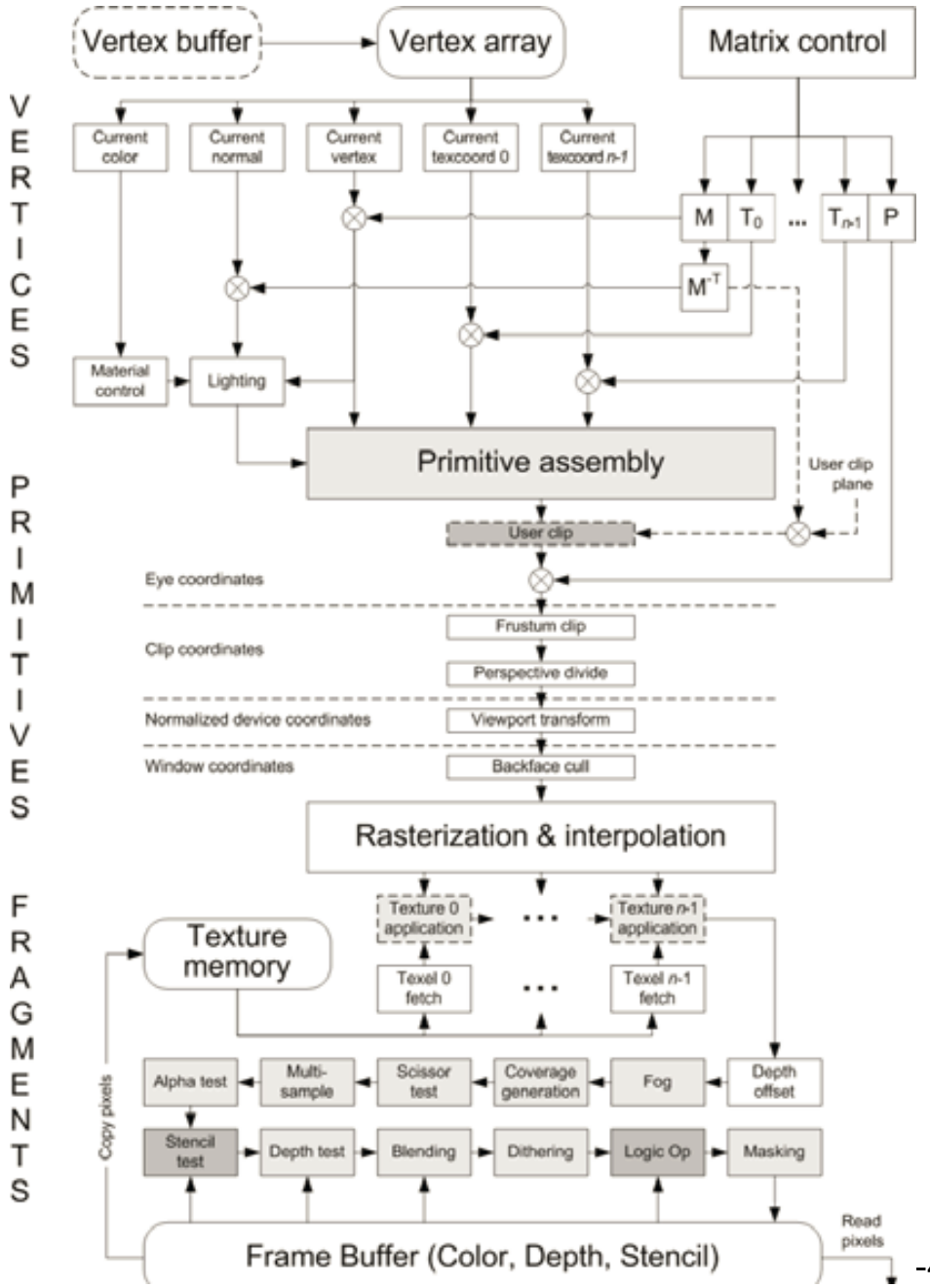

```
struct {
  int depth;
  byte r,g,b,a;
} pixel;
```

Protočni sustav za OpenGL ES (ugrađene sustave)

– vrhovi

– primitive

– fragmenti



Osnove GLUT-a

struktura aplikacije

- konfiguracija i otvaranje prozora
- inicijalizacija OpenGL stanja
  - npr. boja pozadine, položaji izvora,
- funkcije povratnog poziva (engl. callback) – kada se dese događaji pozivaju se funkcije (osvježi prozor…)
  - prikaz – osvježavanje prikaza
  - promjena veličine prozora
  - ulazne naprave: miš, tipkovnica - rukovanje događajima (engl. events)
  - animacija
- ulazak u glavnu petlju procesiranja događaja
  - aplikacija prati događaje i rukuje (poziva funkcije ovisno o događaju)

primjer glavnog programa:

```
void main (int argc, char** argv)
{
  glutInitDisplayMode (GLUT_RGB | GLUT_DOUBLE);    // konfiguracija i otvaranje prozora
  glutInitWindowSize (200, 300);
  glutInitWindowPosition (100, 100);
  glutCreateWindow (“Moj prozor”);

  glClearColor( 0.0, 0.0, 0.0, 1.0 );   // inicijalizacija OpenGL stanja
  glClearDepth( 1.0 );                  // inicijalizacija početnih stanja je često u potprogramu init();
  glEnable( GL_LIGHT0 );
  glEnable( GL_LIGHTING );
  glEnable( GL_DEPTH_TEST );


  glutDisplayFunc (crtaj);              // funkcije povratnog poziva   --  callback
  glutReshapeFunc (resize);
  glutKeyboardFunc (tipkovnica);
  glutIdleFunc (idle);

  glutMainLoop();                         // glavna petlja procesiranja događaja
}
```

neke od funkcija povratnog poziva koje GLUT podržava:

- korisnik treba napisati ove funkcije

poziva se kada:

```
glutDisplayFunc (crtaj);  – treba osvježiti slikovne elemente
glutReshapeFunc (); – se promijeni veličina prozora
glutKeyboardFunc (tipkovnica); – je pritisnuta tipka na tipkovnici
glutMouseFunc (mojmis); - je pritisnuta tipka na mišu
glutMotionFunc (); - je pritisnuta tipka na mišu i pomičemo miša
glutPassiveMouseFunc (); - pomičemo miša neovisno o tipkama miša
glutIdleFunc (); – kada se ništa drugo ne dešava – korisno u animaciji
```

primjer:

```
void crtaj (void)
{
  glClear (GL_COLOR_BUFFER_BIT);          // brisanje i obično glavno iscrtavanje
  glBegin (GL_TRIANGLES);
              glColor3ub(255, 0, 0);  glVertex3f(-1.0, 0.0, 0.0);
              glColor3ub(0, 0, 0);     glVertex3f(0.0, 1.0, 0.0);
              glColor3ub(100, 0, 0);  glVertex3f(0.0, 0.0, 1.0);
  glEnd ();
  glutSwapBuffers ();                              // ako je jedan spremnik onda je glFlush();
}
```

GLUT

pri radu na računalu u okruženju s prozorima (MS Windows, X window Unix, Mac OS):

generiramo niz događaja (event):

- klik mišem, pomak miša
- rad na tipkovnici
- prekrivanje prozora, pomicanje prozora …

generirani događaji prosljeđuju se (window manager) prozoru u kojem su generirani

GLUT – ima niz funkcija povratnog poziva koje reagiraju na događaje

- tri tipa funkcija: window, manu, global (koje su vezane uz prozor, izbornike, globalne)

glutDisplayFunc (crtaj); // mora biti registrirana prije prikaza prozora

- glavna funkcija povratnog poziva (callback) za iscrtavanje
- može biti aktivirano eksplicitno glutPostRedisplay(); kada programer pozove, npr: animacija ili implicitno kao rezultat pomicanja/otklanjanja/zaklanjanja prozora.

glutReshapeFunc (); // ako promijenimo veličinu prozora – što je potrebno napraviti

Interaktivni rad - GLUT funkcije:

- kada je pritisnuta neka tipka tipkovnice poziva se funkcija:

```
glutKeyboardFunc (tipkovnica);

void tipkovnica (unsigned char tipka, int x, int y)    // za tekući prozor pritisak tipke generira
{                                    //  ASSCI znak – pokreće funkciju – prosljeđuje znak i x, y miša
 switch (tipka) {
 case ‘r’ : case ‘R’ :
              glColor3f (1,0,0);        // promijenimo stanje – aktivna boja prikaza je crvena
              glutPostRedisplay();
              break;
 case 27 :  exit (0);
              break;
  }
}
```

- pritisnuta tipka miša:

```
void mojmis (int button, int state, int x, int y)
{
 if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
      {…}
else if (button == GLUT_RIGHT_BUTTON && state == GLUT_DOWN)
      {…}
}
```

Animacija - GLUT funkcije:

- potrebno je osvježavati prikaz za svaki dt pa se zato poziva glutPostRedisplay();

```
glutIdleFunc (idle);

void idle (void)
{
   t += dt;
   glutPostRedisplay(); // u ovom slučaju moramo pozvati ovu funkciju jer se inače neće osvježiti
}
```

OpenGL

Zbog prenosivosti na razne platforme OpenGL ima definirane svoje tipove podataka npr. GLfloat,

ima i specifične npr. za boju GLclumpf 0.0f – 1.1f (u stvari to je i/256, i=0..256)

Grafičko korisničko sučelje

Fizičke naprave

Stisnuta tipka(e)
Miš – pozicija
- promjena, pomak
- klik
3D podaci …

Osluškivanje procesa (*notify process*)

Događaji (*event*)
- kontekstno ovisne akcije
ovisno u kojem prozoru,
na kojem mjestu
i kada (poslije/prije neke akcije)

Podjela zaslona (*layout*), **logički elementi sučelja**
- padajući izbornici (*menu*)
- panel
- canvas

Pokretanje funkcija (*callback*)
- npr. rotacijom u sceni
možemo upravljati s
tipkovnice ili mišem