Jednoprocesorski sustavi

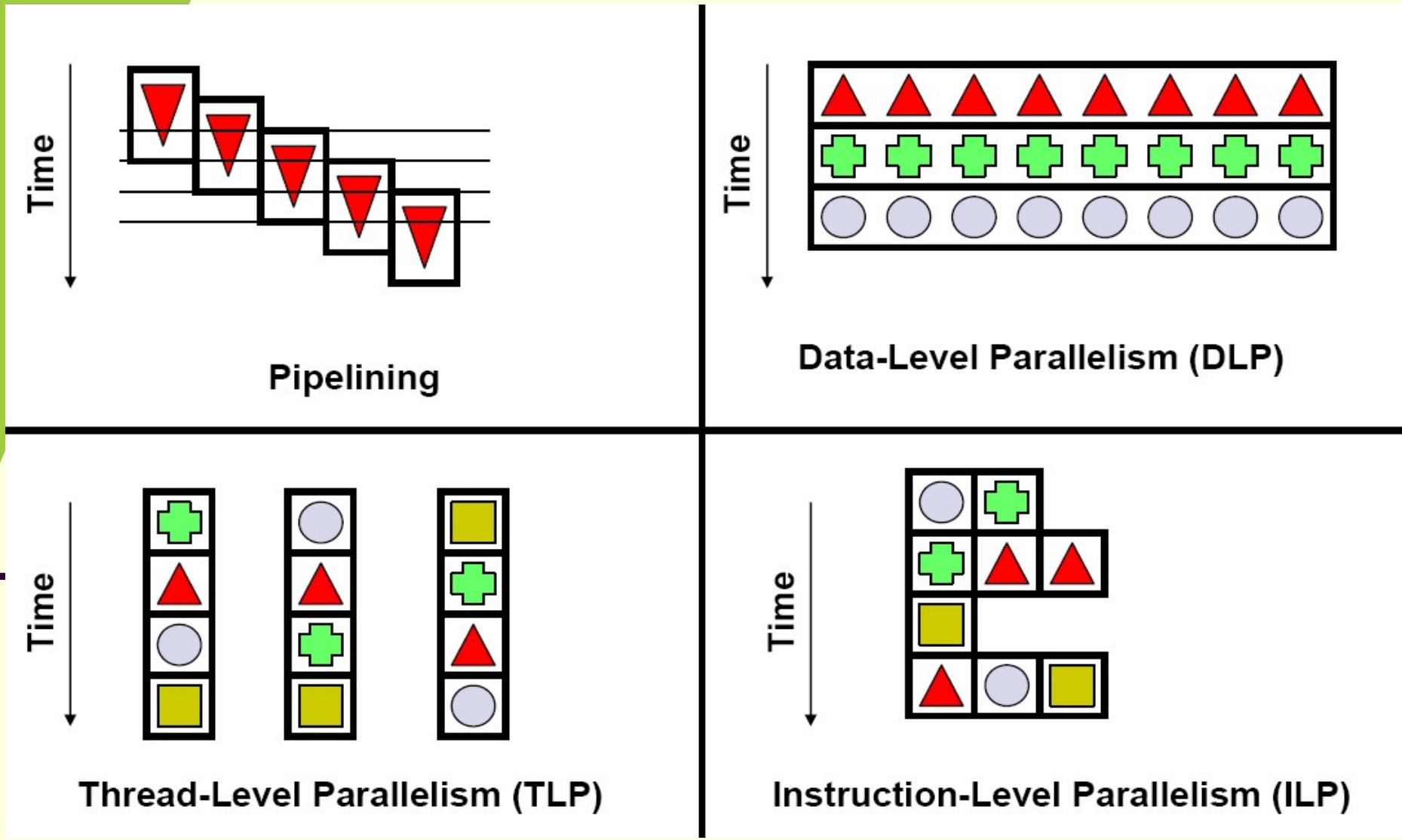
Problemi sa skaliranje uniprocesorske arhitekture:

- Disipacija snage
- Efikasnost
- Kompleksnost
- Kašnjenje u interkonekcijama
- Usporen rast u performansama
- ILP – ne može više dodatno povećavati performanse

Multimedija na višejezgrenim arhitekturama

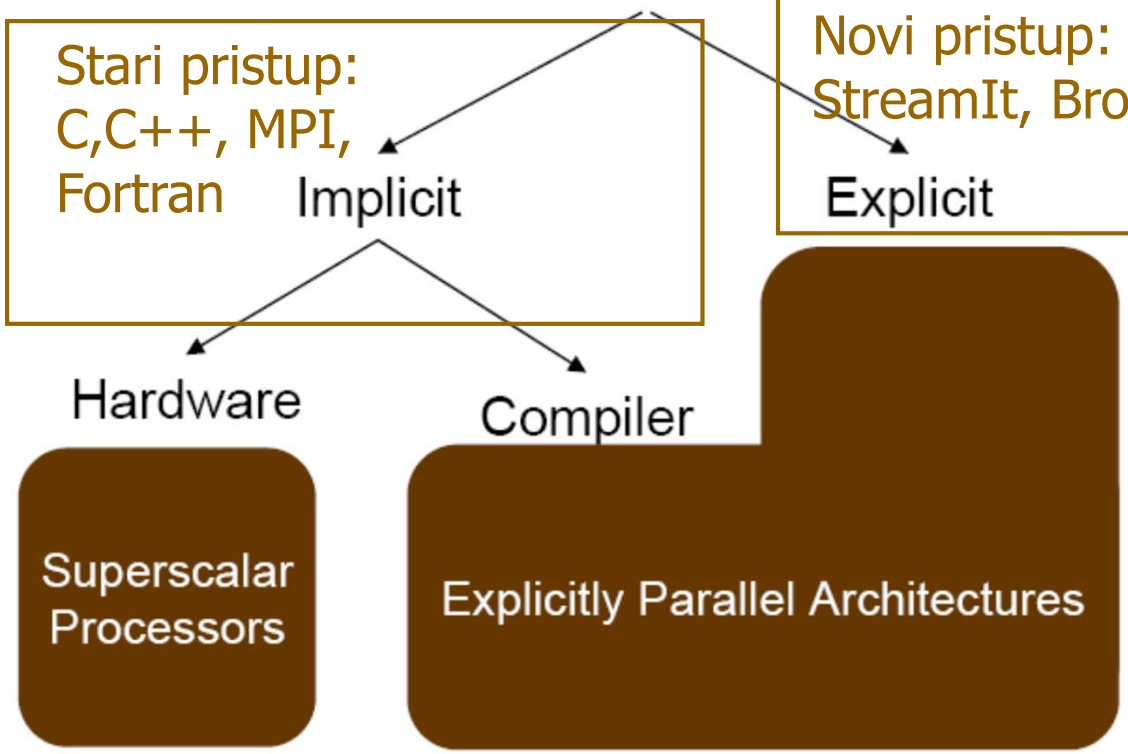
- PARALELIZAM U MM APLIKACIJAMA !!!
- Podatkovni paralelizam DLP (data level)
- Paralelizam među zadatcima TLP (task, thread level)
- Protočni paralelizam PLP (pipeline level)
- Instrukcijski paralelizam ILP (instruction level)

Tipovi paralelizma



Paralelizam u MM aplikacijama

Kako ga opisati i iskoristiti



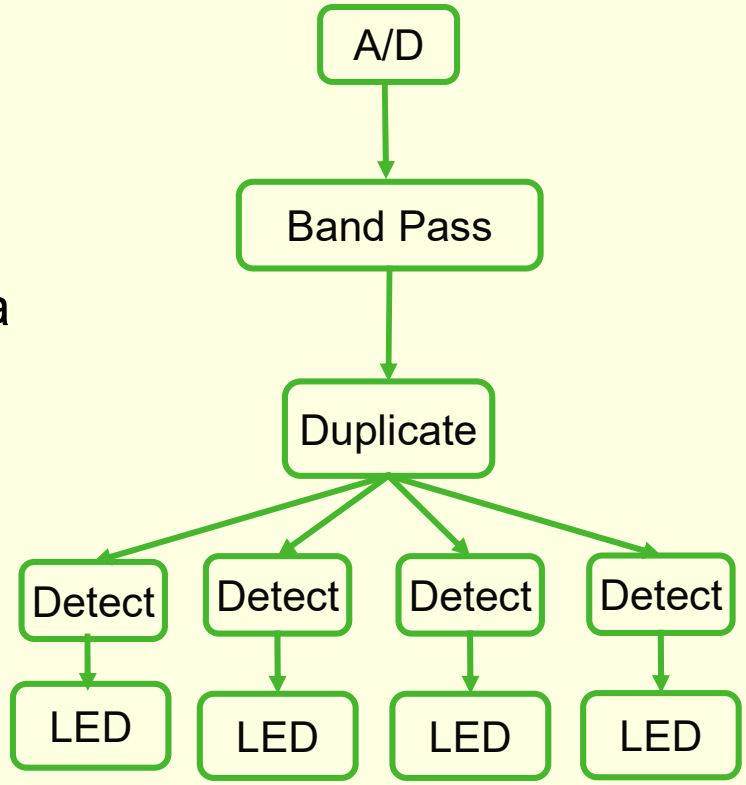
Stari pristup:
- Neprirodno
- Neprenosivost
- Verifikacija!

Novi pristup:
- Prirodno
- Prenosivost
- Verifikacija

http://cag.csail.mit.edu/streamit

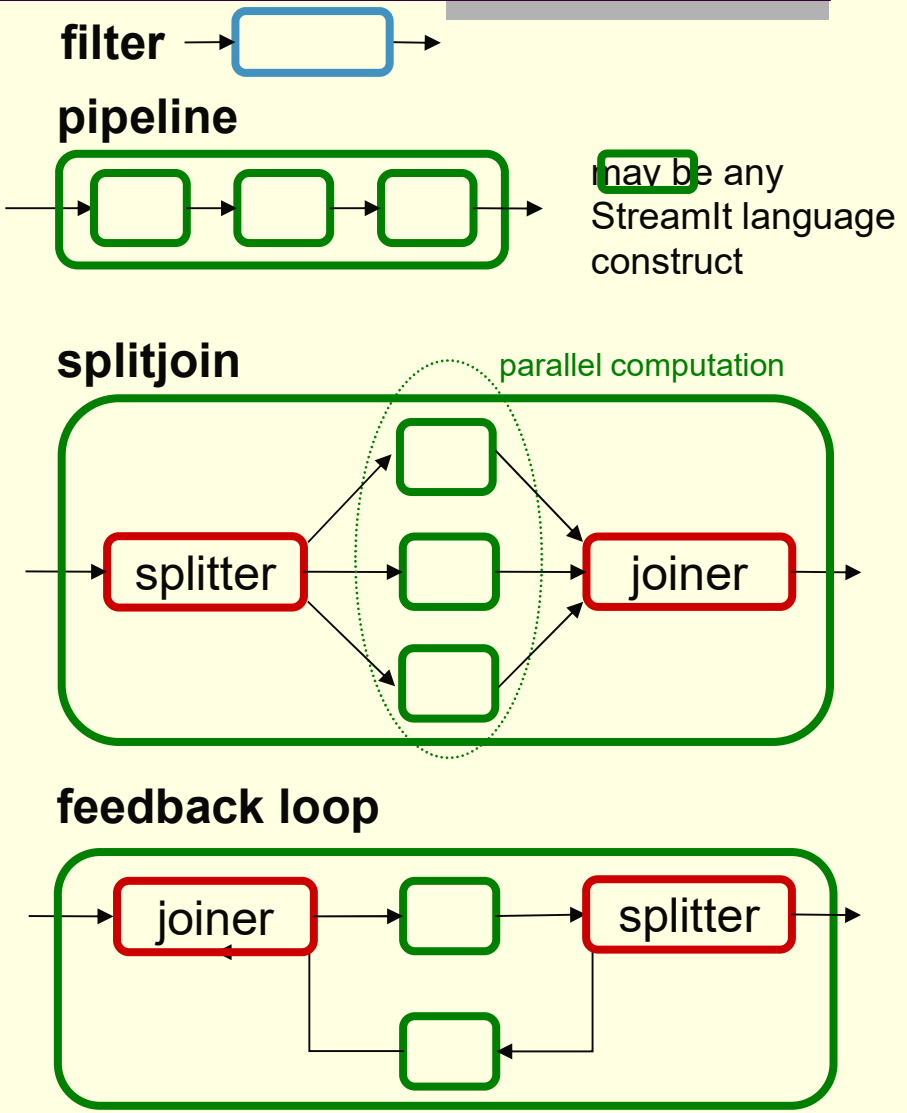
StreamIt (http://cag.csail.mit.edu/streamit)

- SDF Model [Lee_87]
- Graf autonomnih aktora
- Aktori imaju lokaliziran adresni prostor
- Komunikacija preko FIFO kanala
- Statički definirana brzina I/O
- Prevodioc određuje redosljed izvođenja



Programski jezik StreamIt

- SDF s dinamičkim proširenjima
- Paralelizam i komunikacija: eksplicitno izraženi u programu
- Neovisnost o arhitekturi
- Portabilnost
- Modularnost (lako slaganje filtera u složenije grafove toka)
- Skalabilnost
- Osnovne konstrukcije:
- Filter (osnovna jedinica)
- Pipeline (sekvenca filtera)
- Splitjoin (scatter-gather)
- Feedback loop

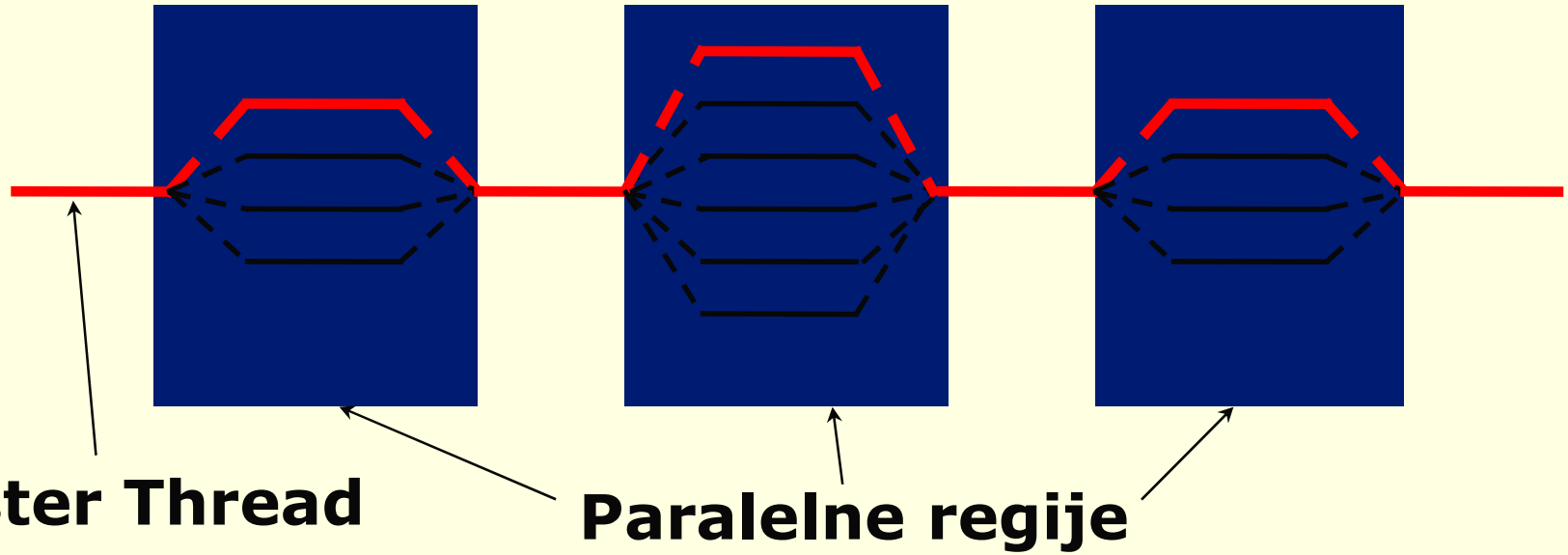


OpenMP

- Skup direktiva i rutina (biblioteka): omogućuje portabilne paralelne aplikacije na SMP arhitekturama
- C/C++, Fortran
- Direktive prevoditelju (compiler directives)
- Data parallelism model i Task parallelism
- Inkrementalni paralelizam
- Kombinira serijski i paralelni kod

OpenMP biblioteka

- Fork-join paralelizam (zasnovan na multithreadingu)
- Master thread pokreće druge threadove
- Inkrementalno dodavanje parlelizma – sekvencijalni program evoluira u paralelni

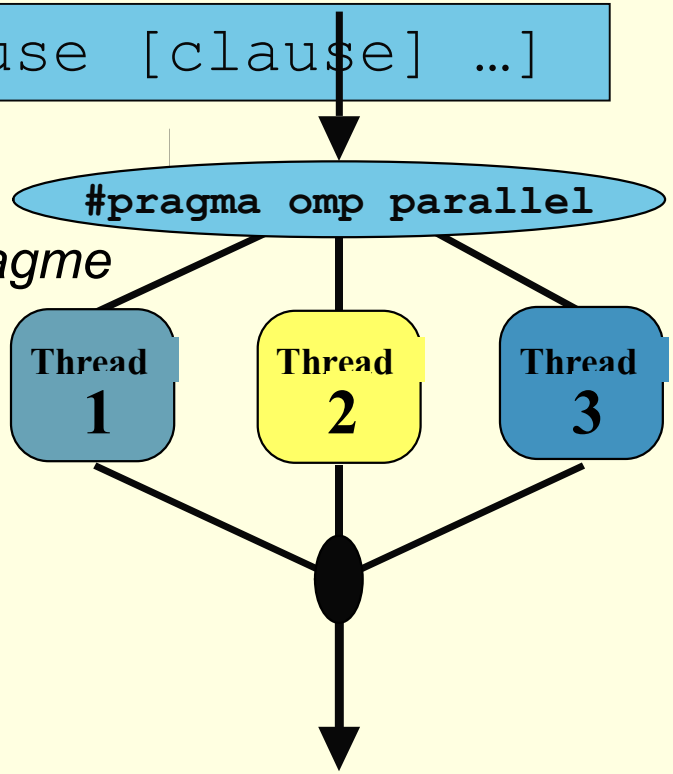


OpenMP biblioteka - sintaksa

Većina OpenMP ključnih riječi je u formi direktiva prevoditelju ili pragma

```
#pragma omp construct [clause [clause] …]
```

- *Paralelne regije*
- *Threadovi se kreiraju pri prolazu pragme*
- *Blokiranje na kraju regije*




```
#pragma omp parallel
{
block
}
```

OpenCL

Heterogeni multiprocesorski sustavi

- CPU, GPU, akcelerator

Model podatkovnog paralelizma (CUDA)

Model paralelizma među zadacima

- Thread -> Work item
- Thread block -> Work group

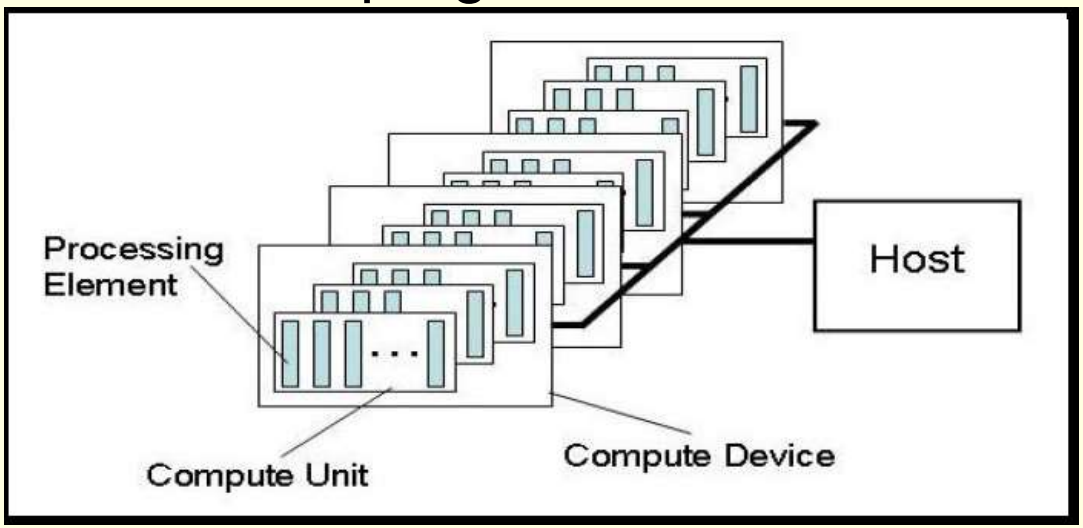
Paralelizam, paralelizam, paralelizam

- Tisuće niti za punu paralelizaciju
- Fokus na “on-chip” memoriju i komunikaciju

OpenCL

Heterogeni multiprocesorski sustavi

- OpenCL device: skup jedne ili više računskih jedinica (compute unit: host + devices (x86 host + GPU device))
- Compute unit: Skup procesnih elemenata (processing elements)
- Processing elements: izvode programski kod



OpenCL izvedbeni model

Kernel (jezgra)

- Osnovna jedinica izvedbenog koda (C funkcija)
- Podatkovno-paralelan, zadatkovno-paralelan

Program

- Skup jezgri i drugih funkcija (dll)

Aplikacija

- Kreira kontekst za upravljanje i izvođenje OpenCL jezgri, razmjenu podataka host-device
- Rep instanci OpenCL jezgri

OpenCL program

Dinamički model prevođenja (OpenGL)

- API pozivi koji omogućuju dinamičku optimizaciju za postojeći OpenCL device

Kreiranje jezgre:

1. OpenCL program spremljen u tekstualnom obliku (učitan u memoriju)
2. Kreiranje programa, API poziv clCreateProgramWithSource()
3. Prevođenje programa za neki OpenCL uređaj, API poziv clBuildProgram()
   - x86: instrukcijski kod
   - GPU: IL (PTX) reprezentacija programa
4. Ekstrakcija jezgre, API poziv clCreateKernel()
5. Prijenos argumenata i “dispatching”, API clSetKernelArg() I clEnqueueNDRangeKernel()