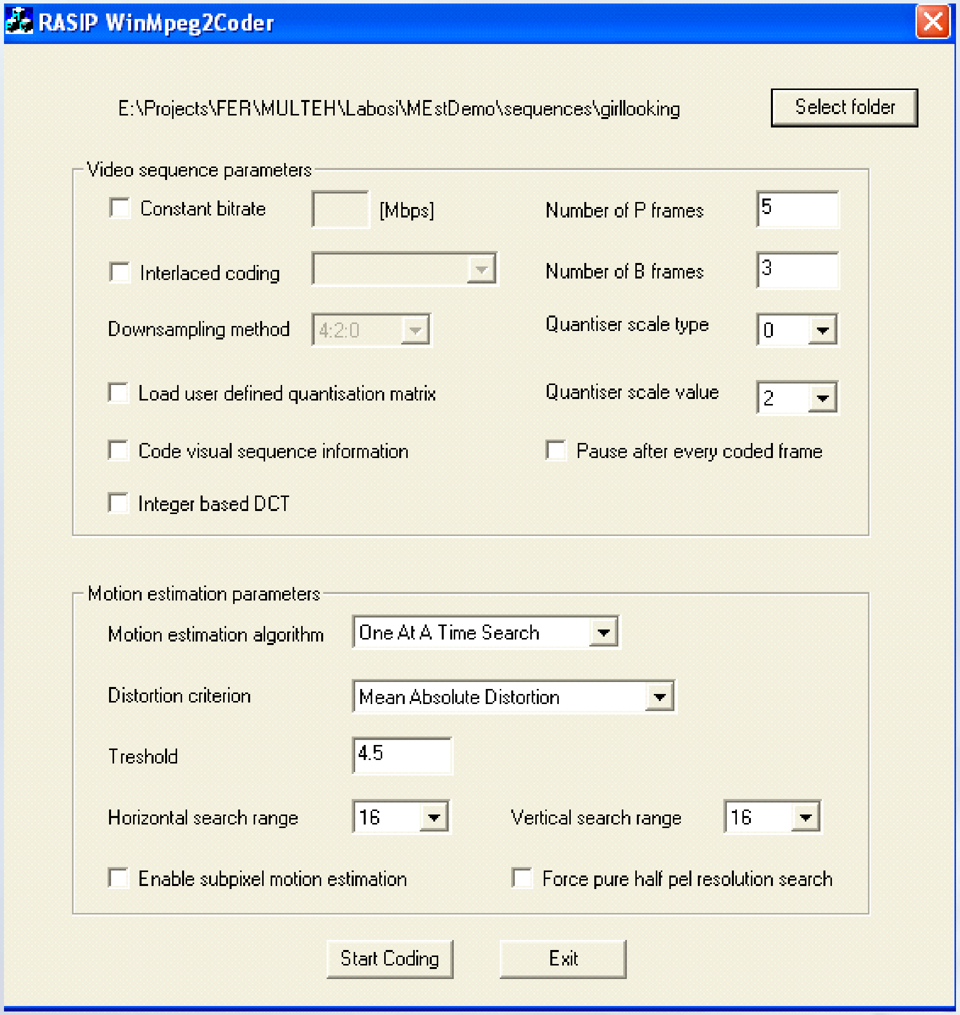
Multimedijske tehnologije

Mr. sc. Josip Knezović (josip.knezovic@fer.hr)

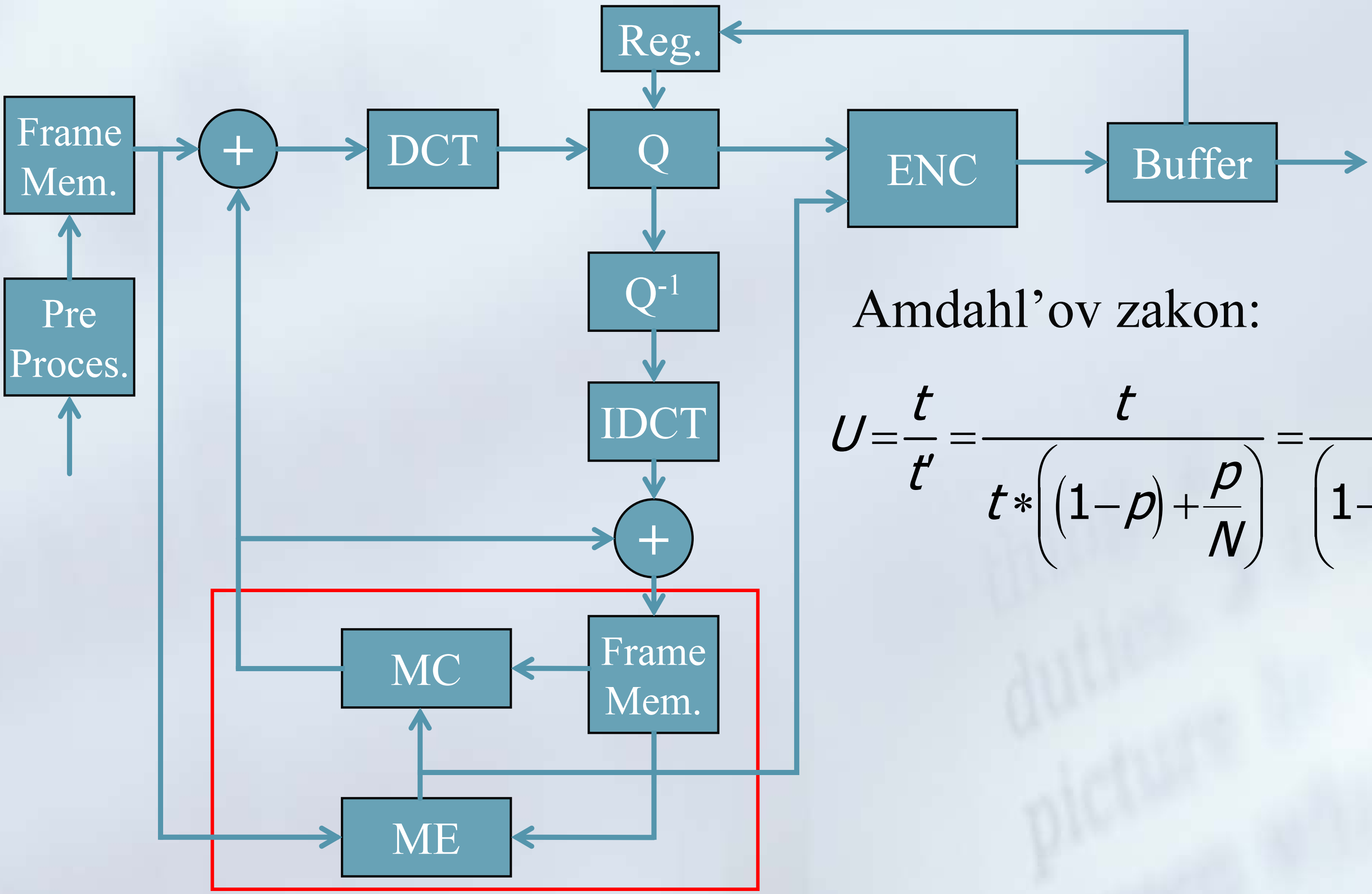
Sadržaj:

1. Lab. vježba - Zadatak 1: MestDemo
2. Lab. vježba - Zadatak 2: Metoda za kodiranje slika slična JPEG normi
3. Nove, višejezgrene arhitekture
4. Programiranje multimedijskih aplikacija za nove, višejezgrene arhitekture

1. Zadatak 1. MestDemo



1. MPEG koder



Amdahl'ov zakon:

$$U = \frac{t}{t'} = \frac{t}{t * \left((1-p) + \frac{p}{N} \right)} = \frac{1}{\left(1 - p + \frac{p}{N} \right)}$$

1. Zadatak 1. MestDemo

- Ulaz: niz bmp slika koje je potrebno kodirati
- Izlaz: kodirani mpg file
- Moguće je podesiti razne parametre kodiranja (kvantizacija, bitrate, broj P i B okvira, tip metode za procjenu pokreta, tip cost funkcije, itd.)
- Kodirana datoteka se može otvoriti i prikazati u bilo kojem playeru (trebala bi ☺)
- Primjer

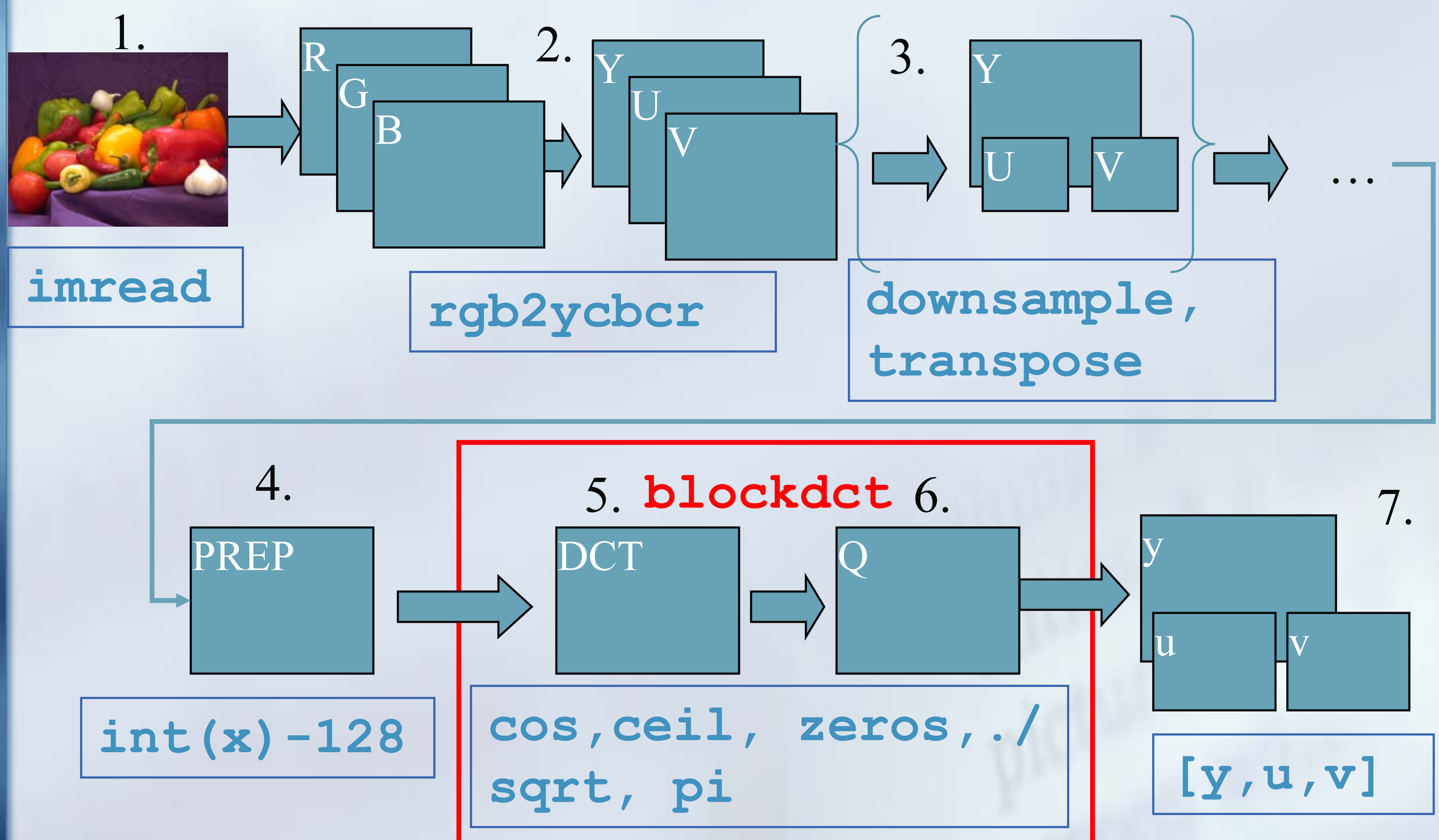
2. Zadatak 2. FCOMP

- Projektirat ćemo metodu za kompresiju slika sa gubitcima sličnu JPEG normi
- Zadatak se izvodi u programskom paketu MATLAB
- Cilj zadatka: stvarni primjer implementacije i optimizacije algoritma
- Koristit ćemo MATLAB funkcije za profiling: `tic, toc, profile`
- `function [y,u,v] = fcomp(x)`

2. FCOMP metoda (JPEG) - MATLAB

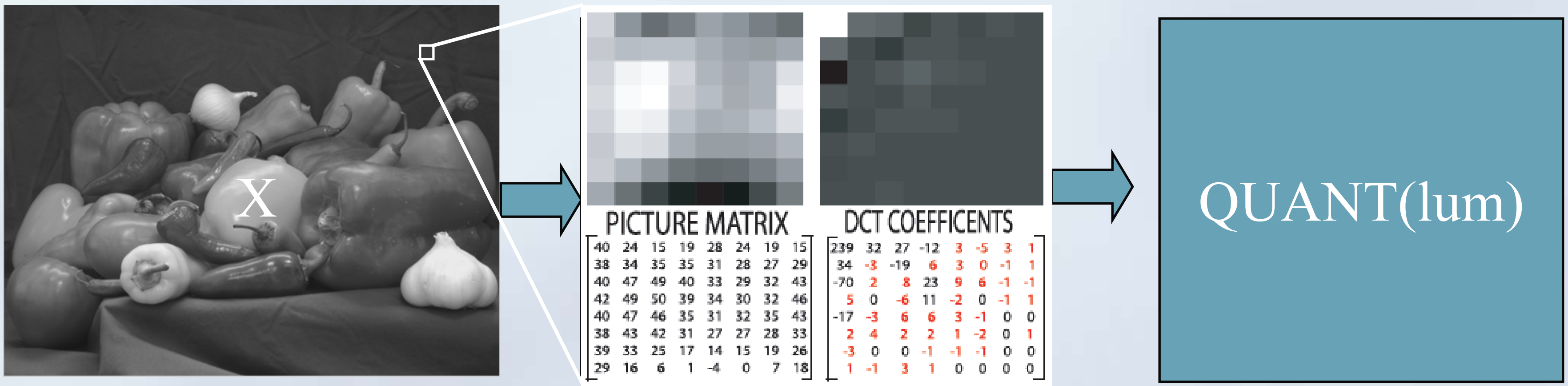
1. Učitavanje BMP slike
2. Transformacija iz RGB u YUV prostor
3. Poduzorkovanje krominantnih komponenti
4. Priprema za DCT (oduzimanje 2^{N-1} od svakog piksela)
5. DCT na 8x8 blokovima
6. Kvantizacija blokova (ovisna o komponenti)
7. Izlaz: tri matrice [y,u,v]

2. FCOMP metoda (JPEG) - MATLAB



2. FCOMP metoda (JPEG) - MATLAB

- `function y = blockdct(x, lum)`



Ulazna matrica x
Tip: lum=1

8x8 blok ulazne matrice
8x8 blok dct koeficijenata
Postupak se ponavlja za svaki 8x8 blok u ulaznoj matrici

2. FCOMP metoda (JPEG) - MATLAB

- Blockdct razbija ulaznu matricu x u 8x8 blokove
- DCT se računa na 8x8 blokovima
- Kvantizacija se radi sa koeficijentima ovisim o tipu matrice

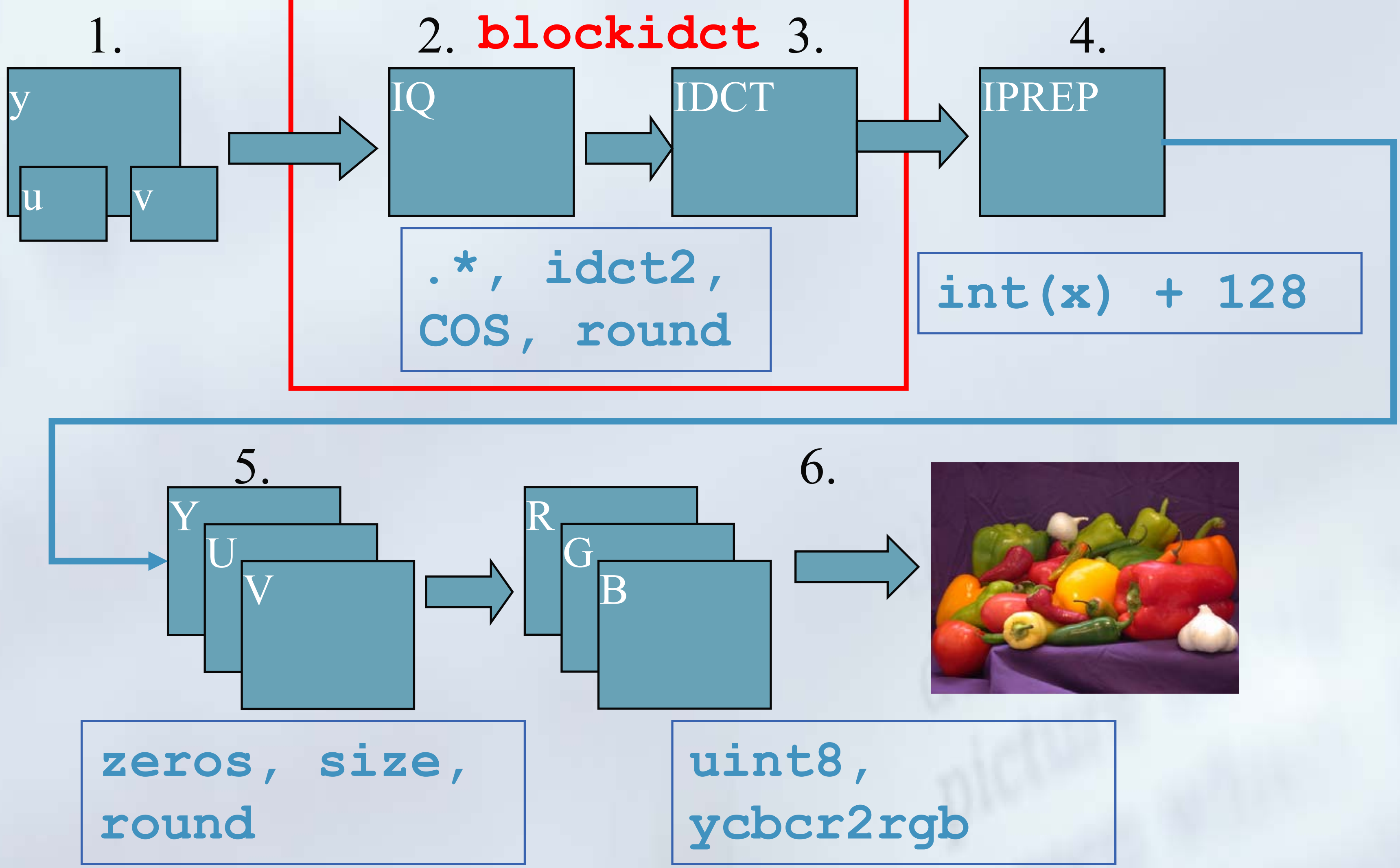
$$qy = \begin{bmatrix} 3 & 2 & 2 & 3 & 5 & 8 & 10 & 12 \\ 2 & 2 & 3 & 4 & 5 & 12 & 12 & 11 \\ 3 & 3 & 3 & 5 & 8 & 11 & 14 & 11 \\ 3 & 3 & 4 & 6 & 10 & 17 & 16 & 12 \\ 4 & 4 & 7 & 11 & 14 & 22 & 21 & 15 \\ 5 & 7 & 11 & 13 & 16 & 21 & 23 & 18 \\ 10 & 13 & 16 & 17 & 21 & 24 & 24 & 20 \\ 14 & 18 & 19 & 20 & 22 & 20 & 21 & 20 \end{bmatrix} \quad qc = \begin{bmatrix} 3 & 4 & 5 & 9 & 20 & 20 & 20 & 20 \\ 4 & 4 & 5 & 13 & 20 & 20 & 20 & 20 \\ 5 & 5 & 11 & 20 & 20 & 20 & 20 & 20 \\ 9 & 13 & 20 & 20 & 20 & 20 & 20 & 20 \\ 20 & 20 & 20 & 20 & 20 & 20 & 20 & 20 \\ 20 & 20 & 20 & 20 & 20 & 20 & 20 & 20 \\ 20 & 20 & 20 & 20 & 20 & 20 & 20 & 20 \\ 20 & 20 & 20 & 20 & 20 & 20 & 20 & 20 \end{bmatrix}$$

2. FEXP - MATLAB

1. Ulaz: tri matrice [y,u,v]
2. Inverzna kvantizacija na blokovima 8x8
3. Inverzna DCT transformacija na blokovima (`function y = blockidct(x,lum)`)
4. Dodavanje 2^{N-1} na svaki piksel
5. Naduzorkovanje krominantnih komponenti
6. Generiranje RGB slike

```
function image=fexp(y,u,v)
```

2. FEXP - MATLAB



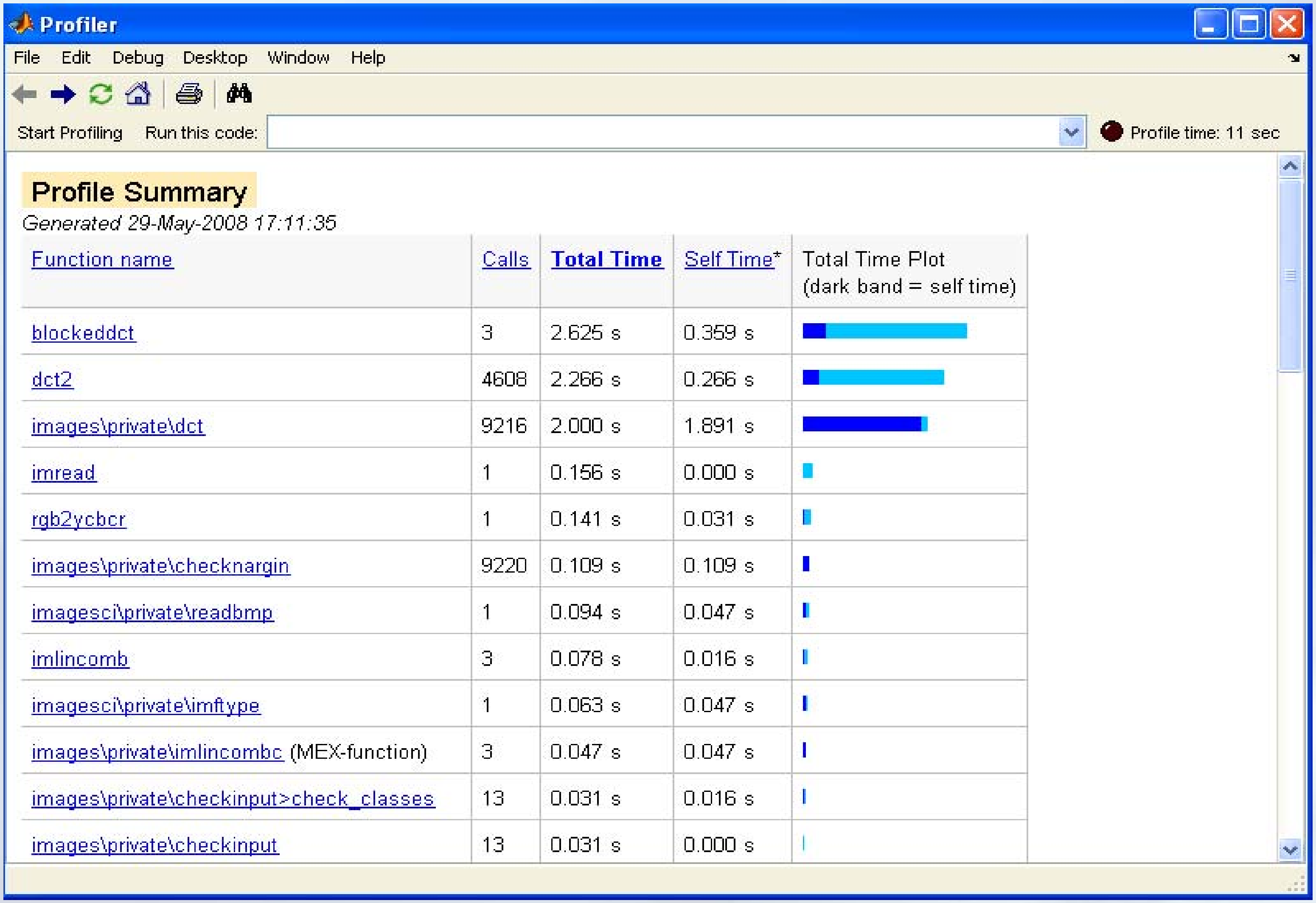
2. FCOMP i FEXP - Profiling

- Potrebno je obaviti profiling izvedenih funkcija
- Profiling obaviti sa: `tic, toc, profile`

```
% MATLAB
>> [y,u,v] = fcomp('peppers.bmp');
>> profile viewer;
```

```
% fcomp.m
function [y,u,v] = fcomp('peppers.bmp')
profile on; % ukljucen profiling
. . .         % tijelo funkcije
[y,u,v] = result;
profile off; %iskljucen profiling
```

2. FCOMP i FEXP - Profiling



Profiler

File Edit Debug Desktop Window Help

Start Profiling Run this code: Profile time: 11 sec

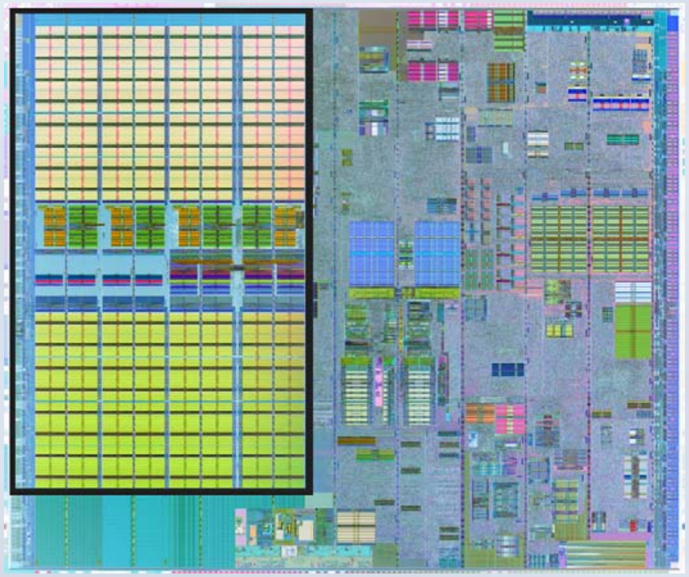
Profile Summary

Generated 29-May-2008 17:11:35

Function name	Calls	**Total Time**	Self Time*	Total Time Plot (dark band = self time)
blockeddct	3	2.625 s	0.359 s	
dct2	4608	2.266 s	0.266 s	
images\private\dct	9216	2.000 s	1.891 s	
imread	1	0.156 s	0.000 s	
rgb2ycbcr	1	0.141 s	0.031 s	
images\private\checknargin	9220	0.109 s	0.109 s	
imagesci\private\readbmp	1	0.094 s	0.047 s	
imlincomb	3	0.078 s	0.016 s	
imagesci\private\imftype	1	0.063 s	0.047 s	
images\private\imlincombc (MEX-function)	3	0.047 s	0.047 s	
images\private\checkinput>check_classes	13	0.031 s	0.016 s	
images\private\checkinput	13	0.031 s	0.000 s	

3. Višejezgrene arhitekture

- Što se desilo sa uniprocesorskom arhitekturom (Primjer: Intel Tejas i Jayhawk – Otkazani!)
- *The Inquirer: The immediate successor to Prescott after it tops out at 5.20GHz will be the "Tejas" core, also produced on a 90 nanometer process and delivering 5.60GHz using a 1066MHz system bus. That's slated to start appearing towards the end of 2004. Tejas will increase in steady increments which appear to be 6GHz, 6.40GHz, 6.80GHz, 7.20GHz, 7.60GHz, 7GHz, 8.40GHz, 8.80GHz and topping out at 9.20GHz. The first Nehalem is supposed to appear at 9.60GHz before Intel succeeds in its goal to produce a 10GHz+ chip, the Nehalem, and using a 1200MHz front side bus.*



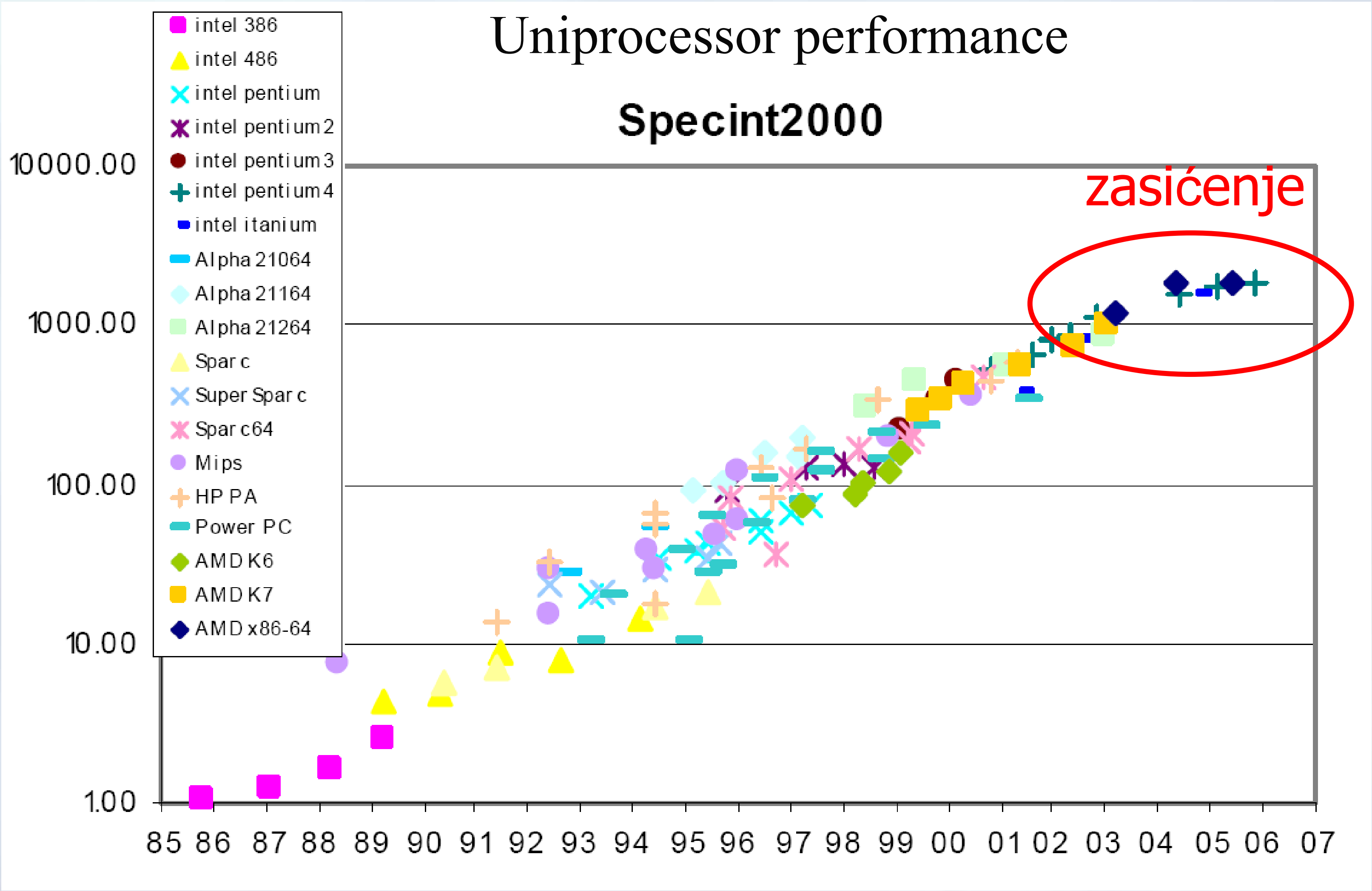
Intel Presscott:

- 30 stupnjeva protočne arhitekture?
- Velika disipacija snage
- Veliki udio L1 i L2 cache memorije u površini sklopa
- Wire delay

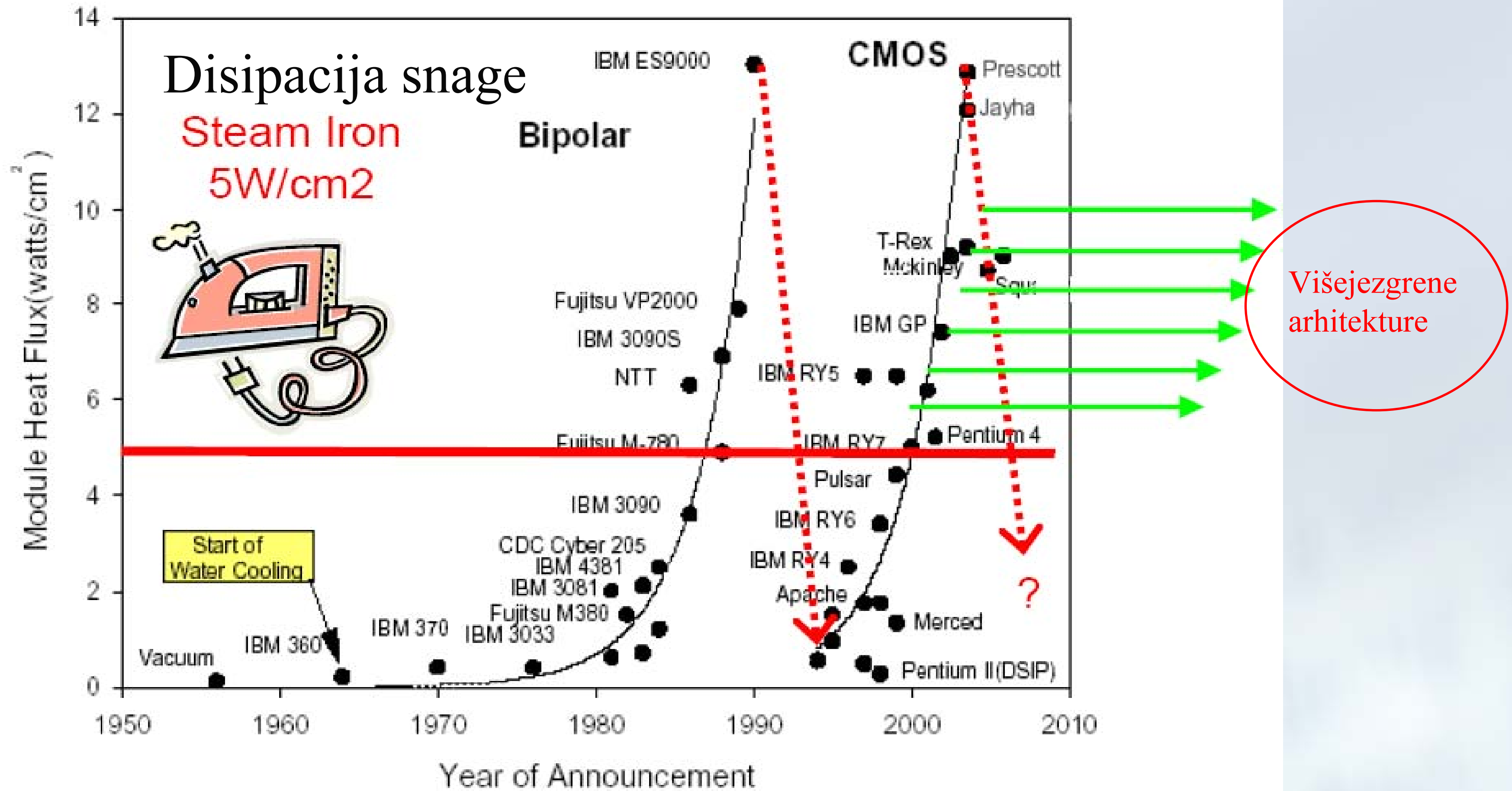
3. Višejezgrene arhitekture

- Problemi sa skaliranje uniprocesorske arhitekture:
  - Disipacija snage
  - Efikasnost
  - Kompleksnost
  - Kašnjenje u interkonekcijama
  - Usporen rast u performansama
  - ILP – ne može više dodatno povećavati performanse

3. Višejezgrene arhitekture

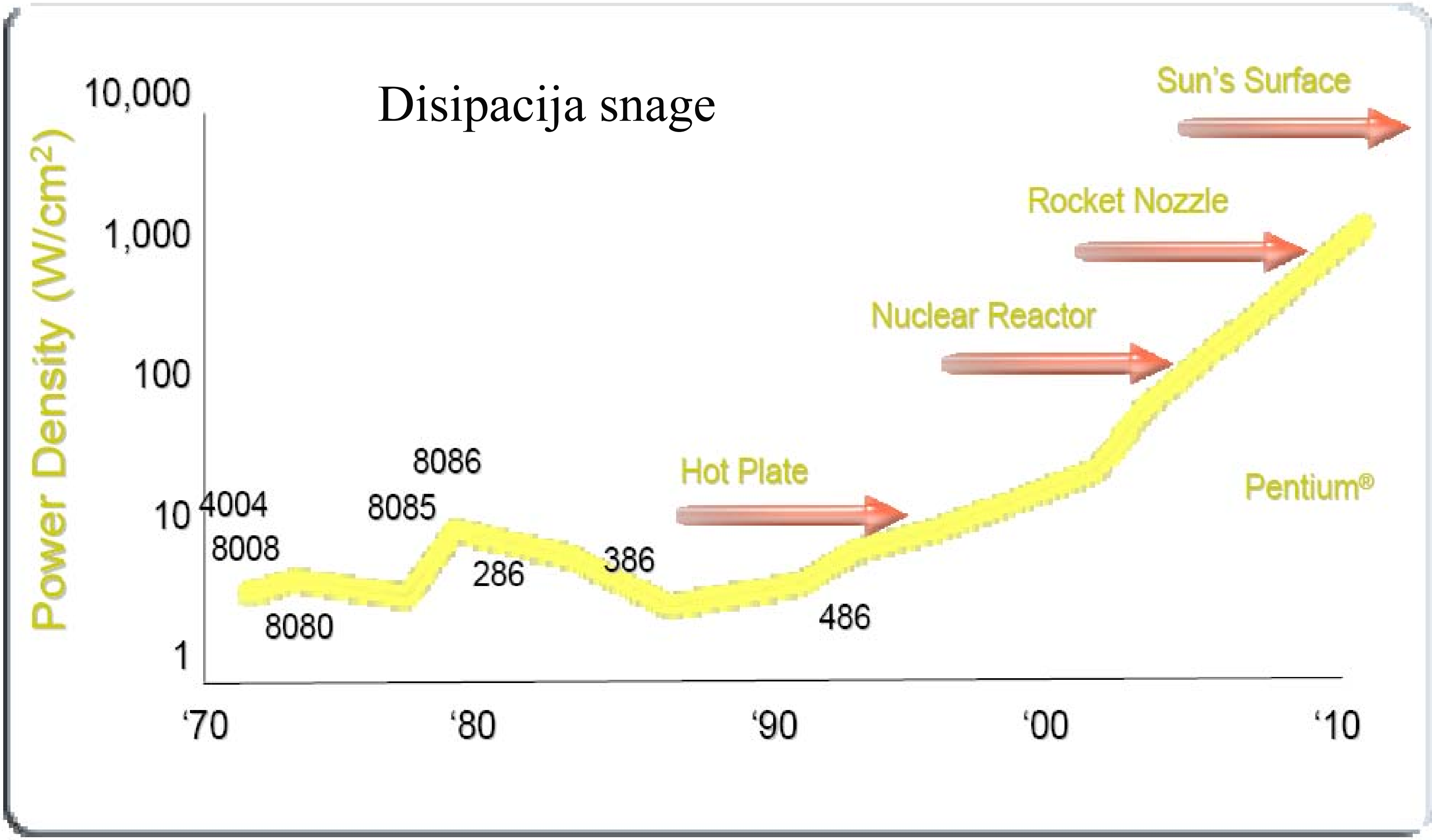


3. Višejezgrene arhitekture



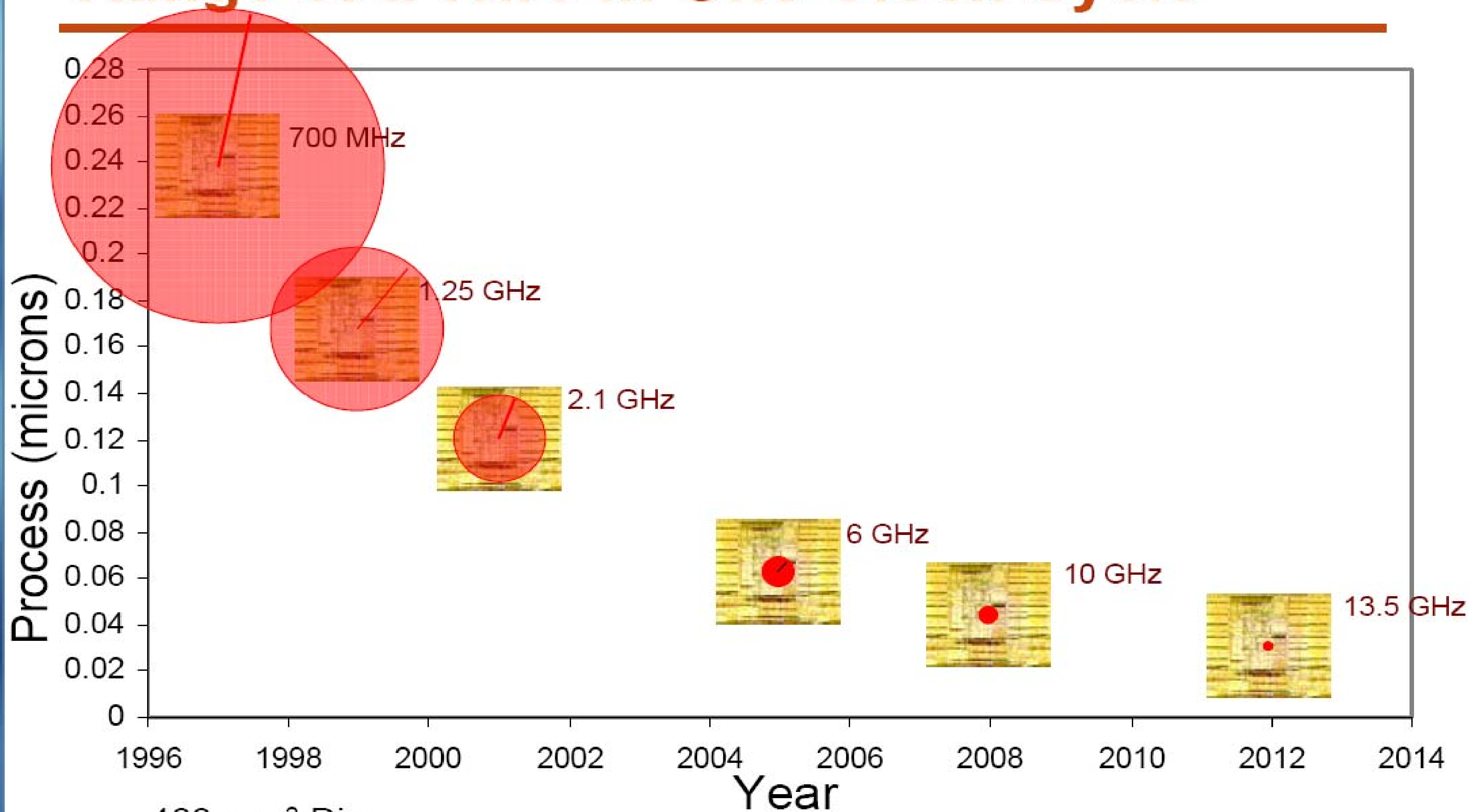
Dr. Michael Perrone – Introduction to the Cell Processor
MIT Class 6.189 Multicore Programming Primer

3. Višejezgrene arhitekture



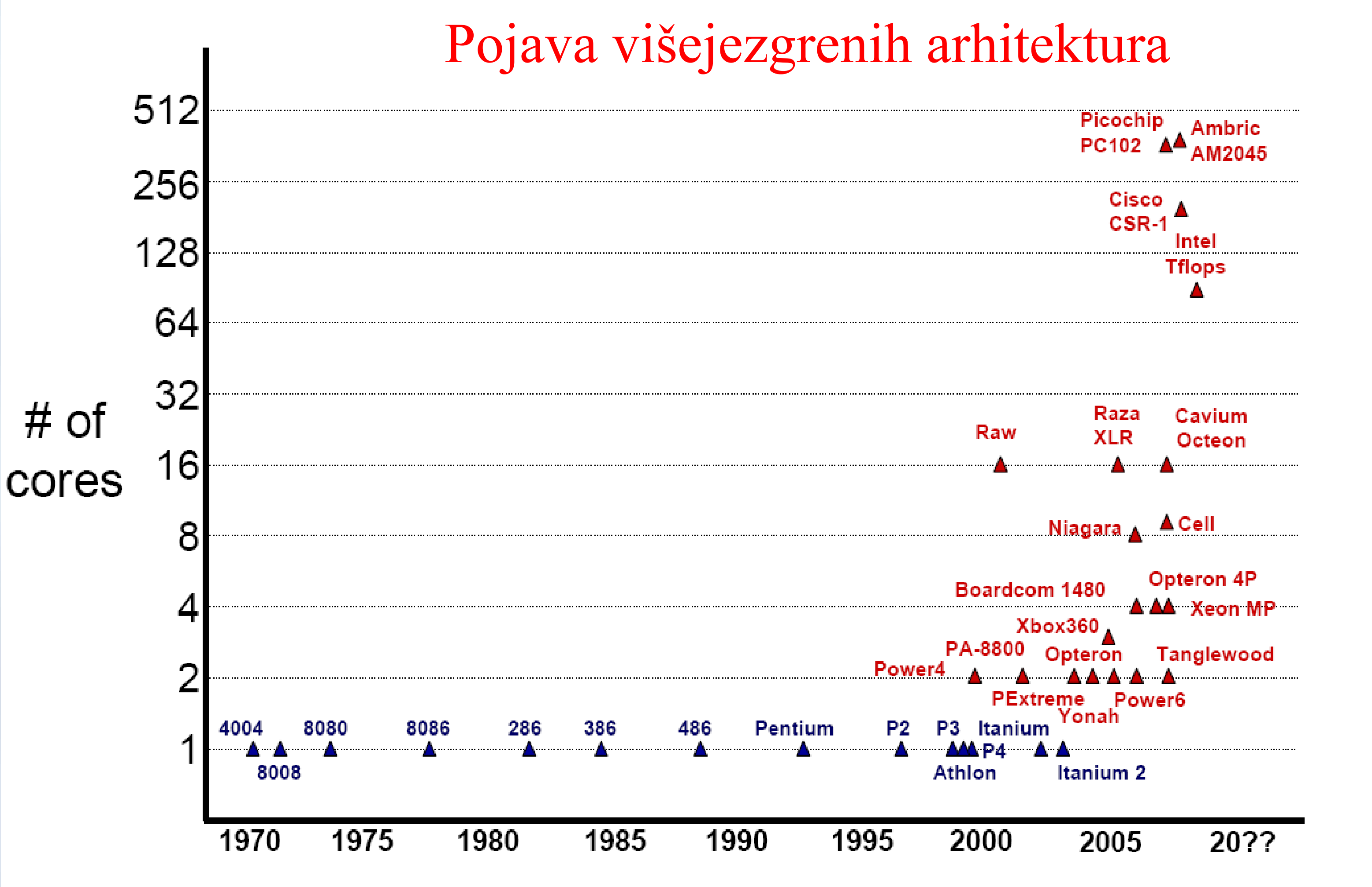
3. Višejezgrene arhitekture

Range of a Wire in One Clock Cycle



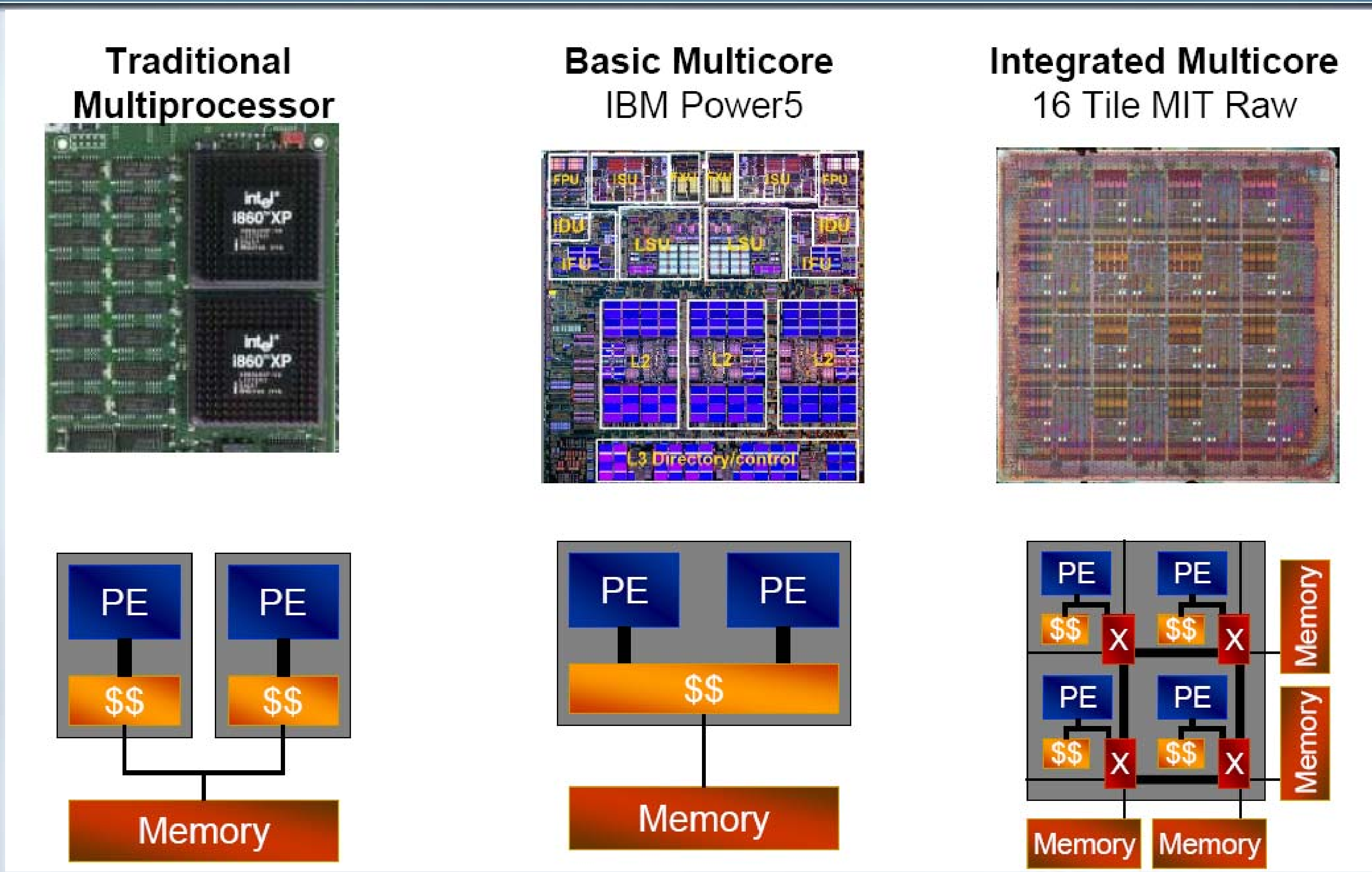
- 400 mm² Die
- From the SIA Roadmap

3. Višejezgrene arhitekture



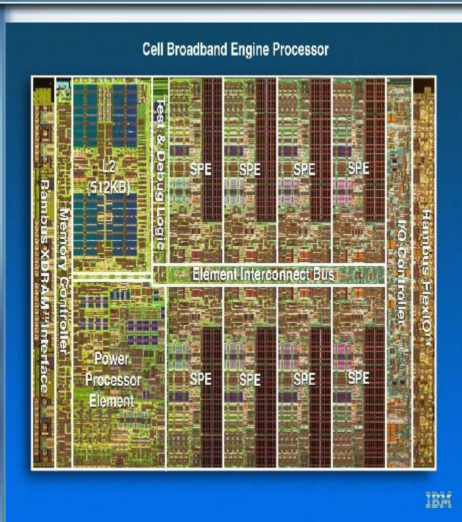
Prof. Saman Amarasinghe, MIT

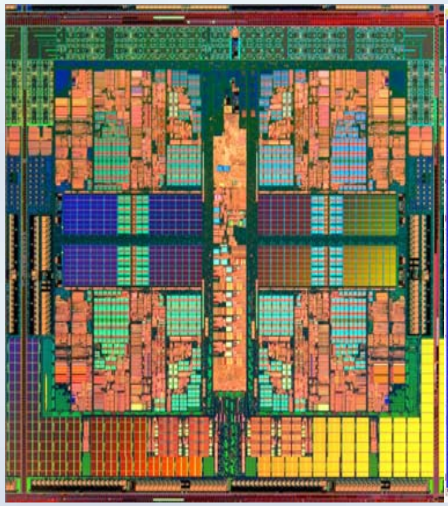
3. Višejezgrene arhitekture

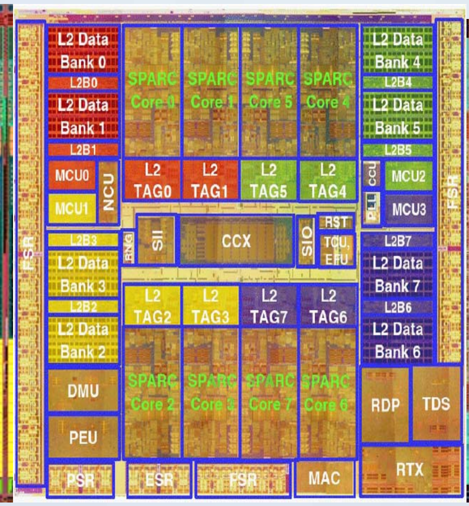


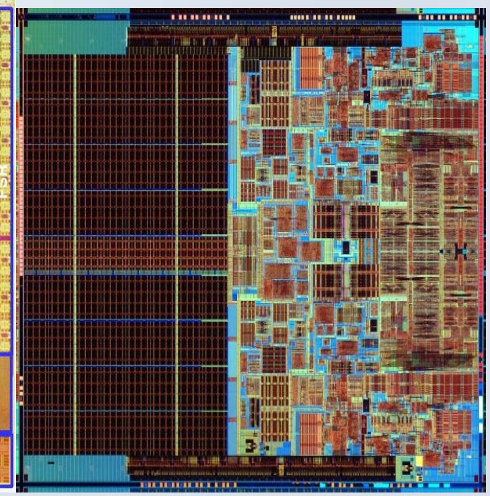
Prof. Saman Amarasinghe, MIT

3. Višejezgrene arhitekture









Cell BE
PS3
Serveri
AMP

Amd Barcelona
Serveri
SMP

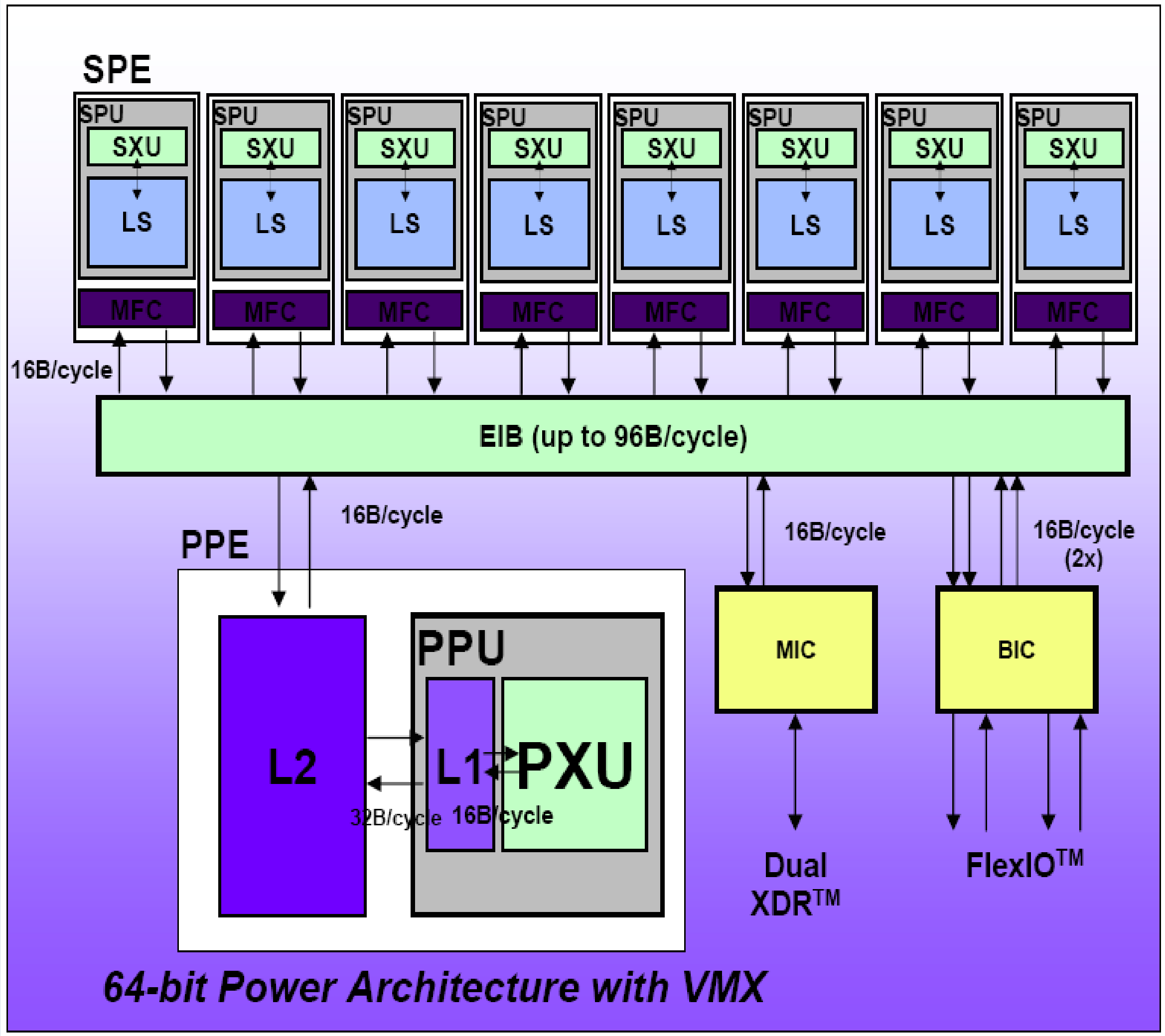
Ultra Sparc T2
Serveri
SMP

Core 2 Duo
Stolna računala
SMP

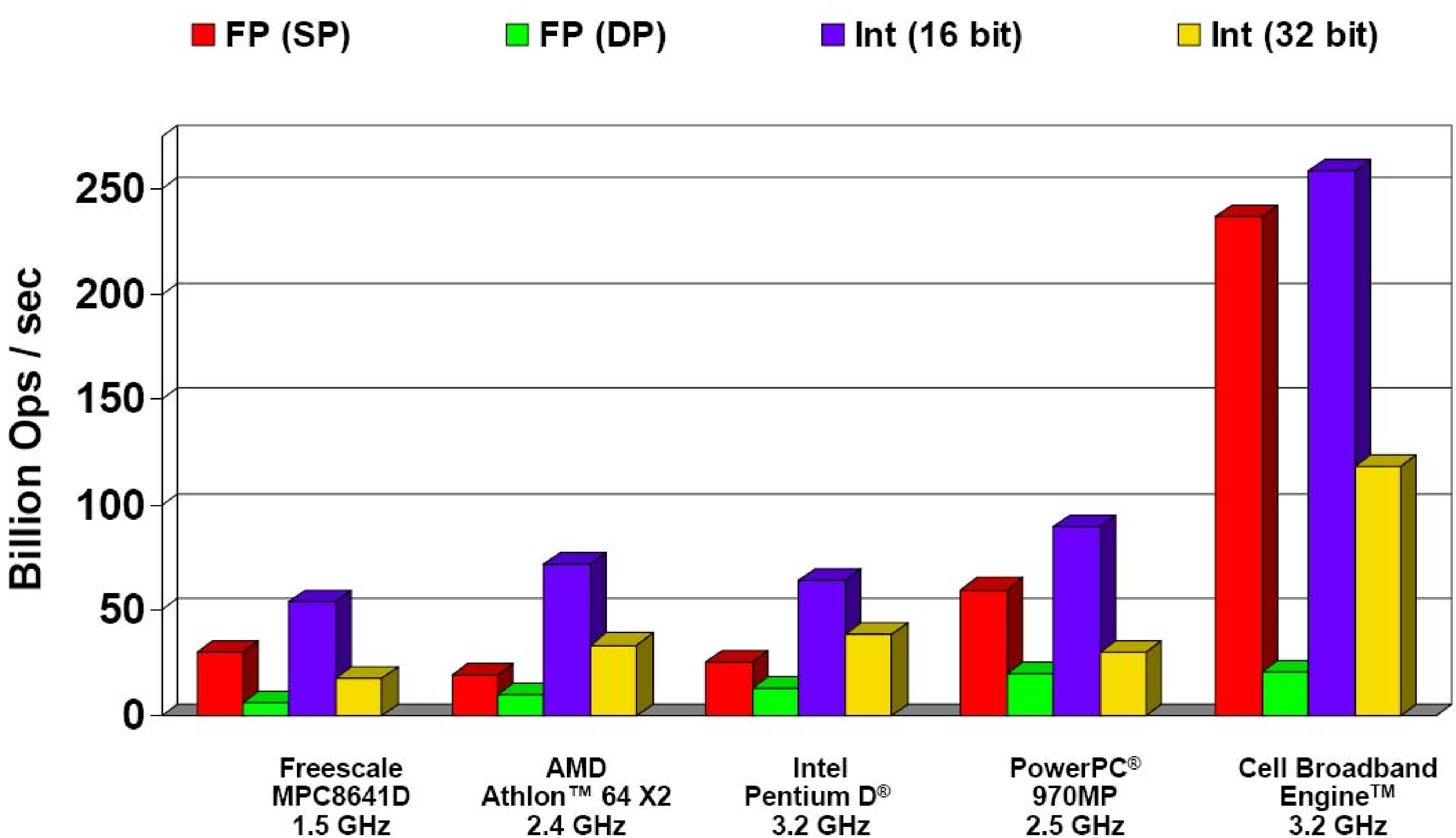
Multimedijske aplikacije – mogu li iskoristiti nove mogućnosti višejezgrenih računalnih sustava?

3. Višejezgrene arhitekture – Cell BE

- PPE
  - Kontrolni zadaci
  - Virtualizacija memorije
  - Operacijski sustav
- SPE
  - Obrada podataka
  - Vektorsko procesiranje
  - Lokalizirana memorija LS mapirana u MS od PPE
  - Brza veza na EIB preko MFC (DMA)
- EIB
  - Podatkovni prsten
  - 300 GB/s



Cell BE – Teoretske vršne performanse



3. Cell BE - Aplikacije

Cell Broadband Engine

- Non-homogeneous coherent multi-Processor
  - Dual-threaded control-plane processor
  - 8 independent data-plane processors
  - Thread-level parallelism
- SIMD processing architecture
  - 128-entry, 128-bit register files
  - Pipelined execution units
  - Branch hint
  - Data-level parallelism
- Rich integer instruction set
  - Word, halfword, byte, bit
  - Boolean
  - Shuffle
  - Rotate, shift, mask
- Single-precision floating point
- Double-precision floating point
- 256KB SPU local stores
  - Asynchronous DMA/main memory interface
  - Channel interface
  - Single-cycle load/store to/from registers
- High-bandwidth internal bus
  - 96 bytes transferred per clock
  - 100+ outstanding transfers supported
- Coherent bus interface
  - Up to 30GB/s out, 25 GB/s in
  - Direct attach of another Cell
  - Can be configured as non-coherent
- Non-coherent bus interface
  - Up to 10GB/s out, 10 GB/s in
- 25+ GB/s XDR memory interface

Accelerated Functions

- Signal processing
- Image processing
- Audio resampling
- Noise generation
- Sound oscillation
- Digital filtering
- Curve and surface evaluation
- FFT
- Matrix mathematics
- Vector mathematics
- Game Physics / Physics simulation
- Video compression / decompression
- Surface subdivision
- Transform-light
- Graphics content creation
- Security encryption / decryption
- Pattern matching
- Language parsing
- TCP/IP offload
- Encoding / decoding
- Parallel processing
- Real time processing
- ...

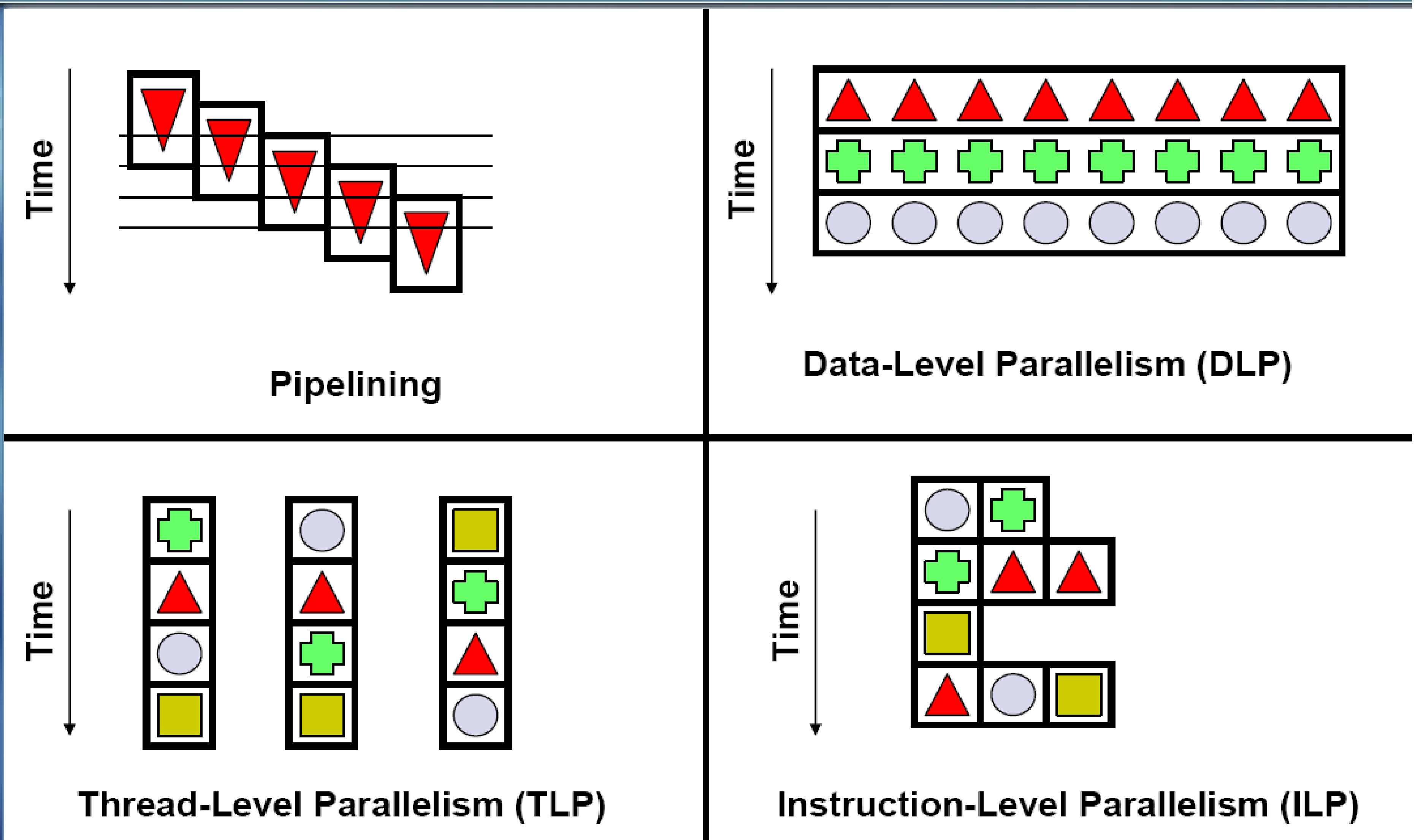
Target Applications

- Medical imaging / visualization
- Drug discovery
- Petroleum reservoir modeling
- Seismic analysis
- Avionics
- Air traffic control systems
- Radar systems
- Sonar systems
- Training simulation
- Targeting
- Defense and security IT
- Surveillance
- Secure communications
- LAN/MAN Routers
- Network processing
- XML and SSL acceleration
- Voice and pattern recognition
- Video conferencing
- Computational chemistry
- Climate modeling
- Data mining and analysis
- Media server
- Digital content creation
- Digital content distribution
- ....

4. Multimedija na višejezgrenim arhitekturama

- PARALELIZAM U MM APLIKACIJAMA !!!
  - Podatkovni paralelizam DLP (dana level)
  - Paralelizam među zadatcima TLP (task, thread level)
  - Protočni paralelizam PLP (pipeline level)
  - Instrukcijski paralelizam ILP (instruction level)
- KAKO GA ISKORISTITI NA VIŠEJEZGRENIM ARHITEKTURAMA
  - Tradicionalni načini (MPI, Fortran, C++, asembler, paralelizirajući prevodioci, …)
  - Novi pristupi (stream programming, …)

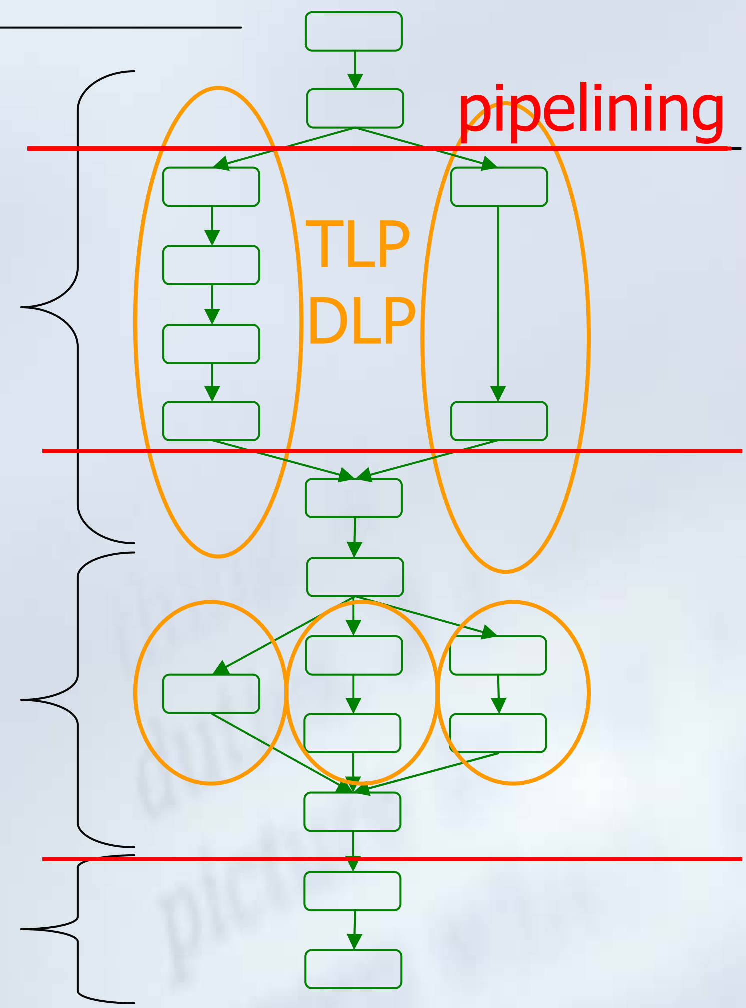
4. Tipovi paralelizma



Prof. Saman Amarasinghe, MIT

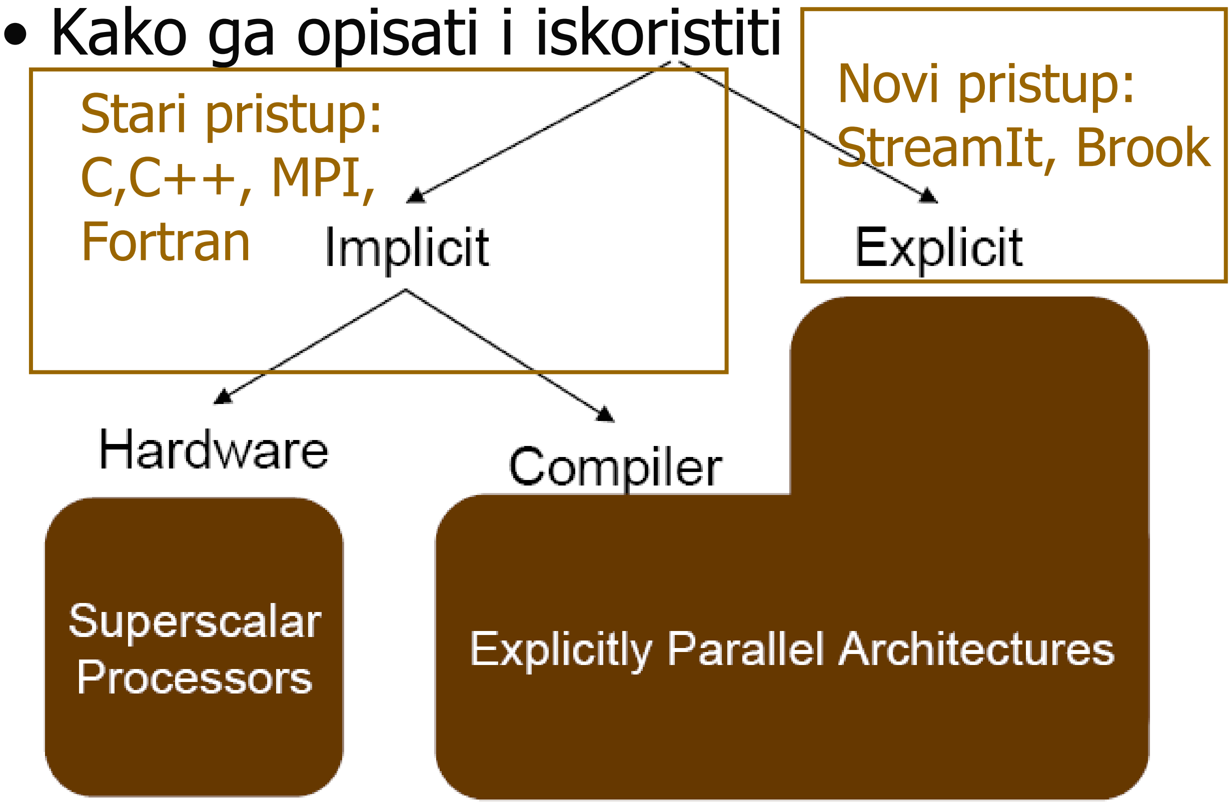
4. Višejezgrene arhitekture – MPEG-2 dekodiranje

- VL dekodiranje
- Prostorno dekodiranje
  - Dekodiranje blokova paralelno sa dekodiranjem vektora pokreta
- Vremensko dekodiranje
  - Dekodiranje komponenti YUV u paraleli
- Pretvorba prostora boja i preslagivanje podataka



4. Paralelizam u MM aplikacijama

- Kako ga opisati i iskoristiti



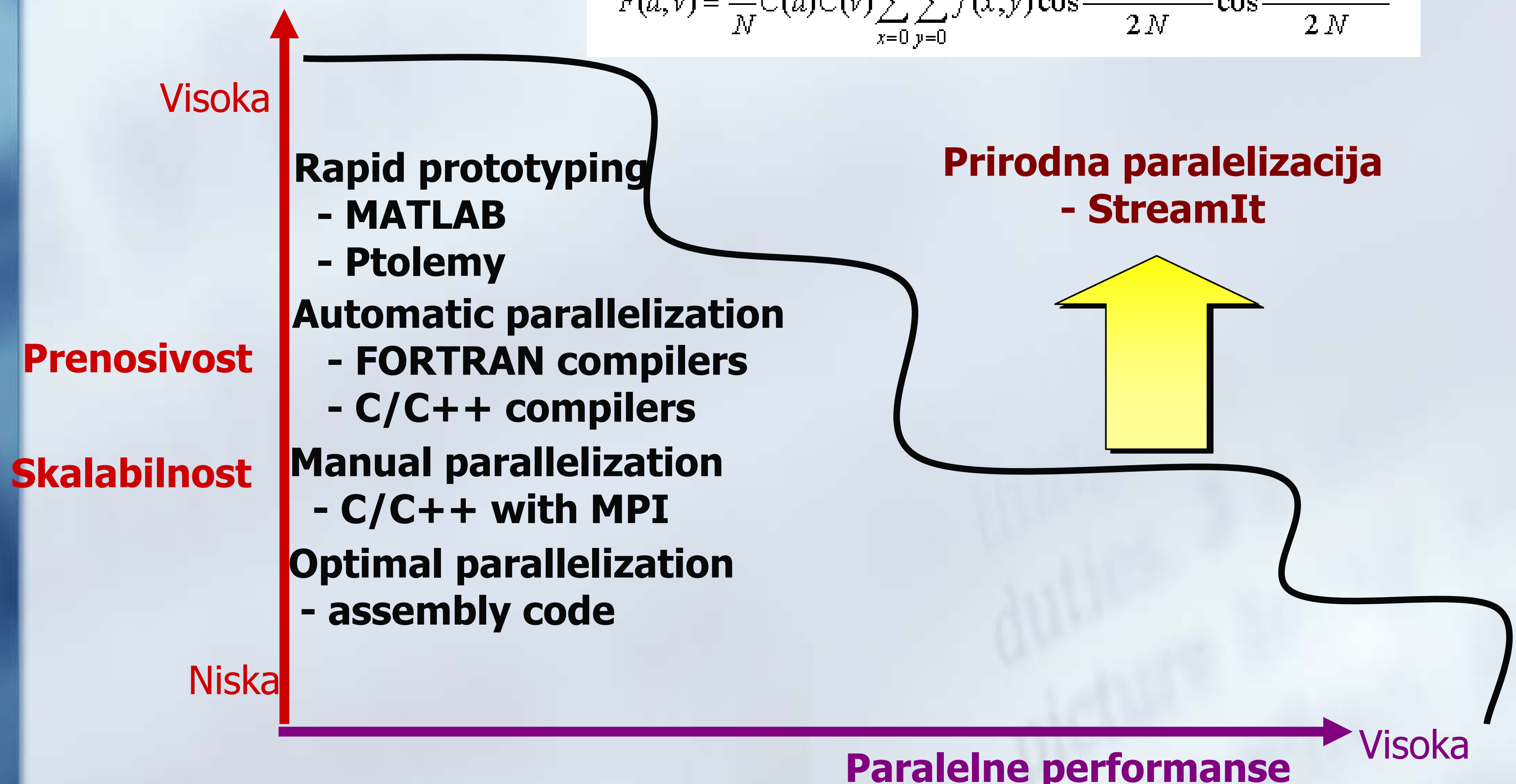
Stari pristup:
- Neprirodno
- Neprenosivost
- Verifikacija!

Novi pristup:
- Prirodno
- Prenosivost
- Verifikacija

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

4. Što učiniti sa paralelizmom u MM

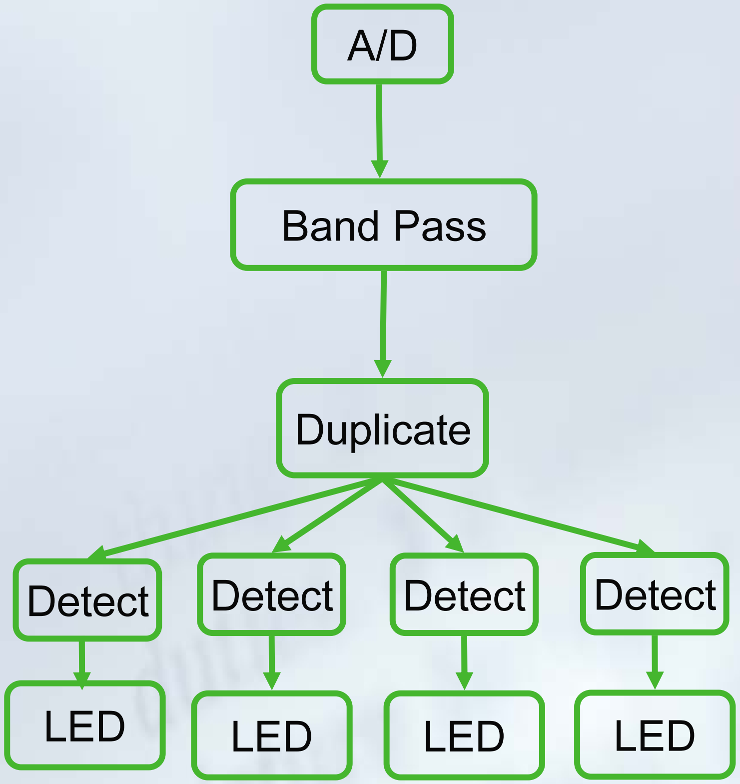
$$F(u,v)=\frac{2}{N}C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N}$$



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

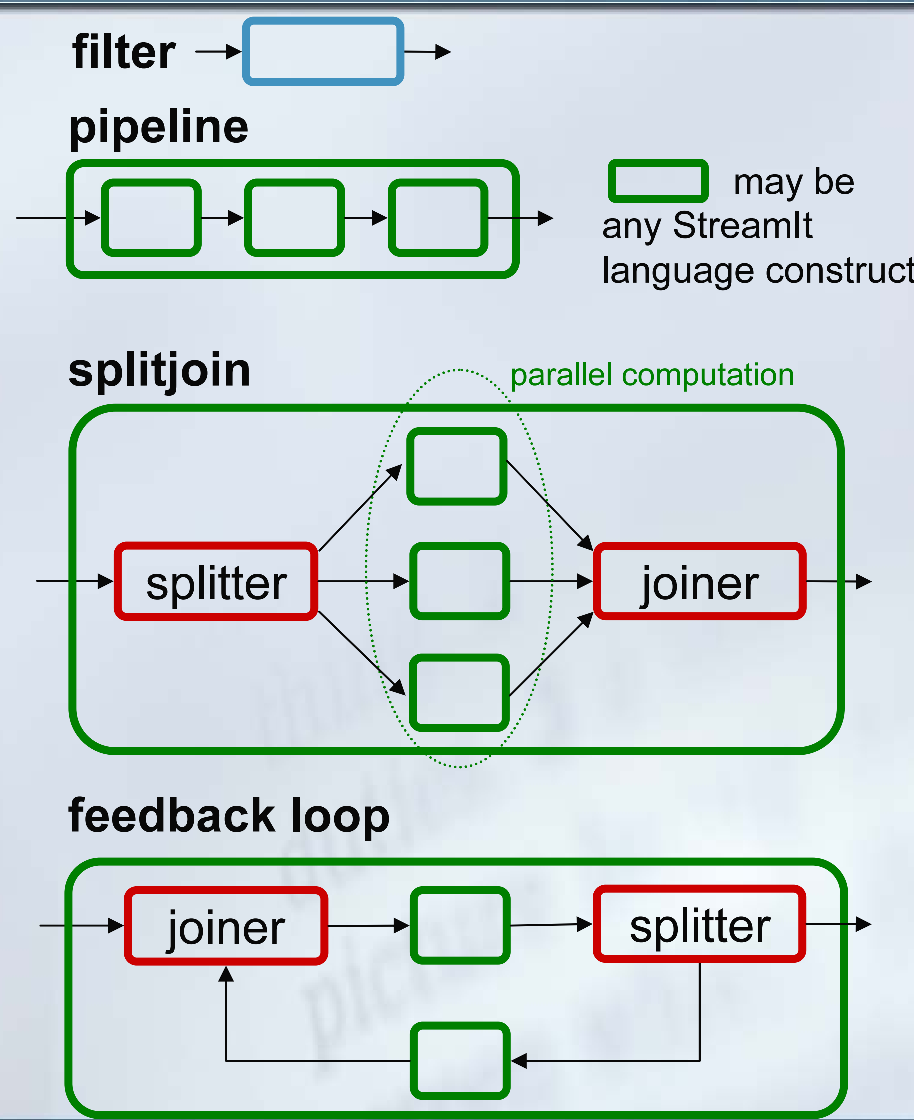
4. StreamIt (http://cag.csail.mit.edu/streamit)

- 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



4. StreamIt programski jezik

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

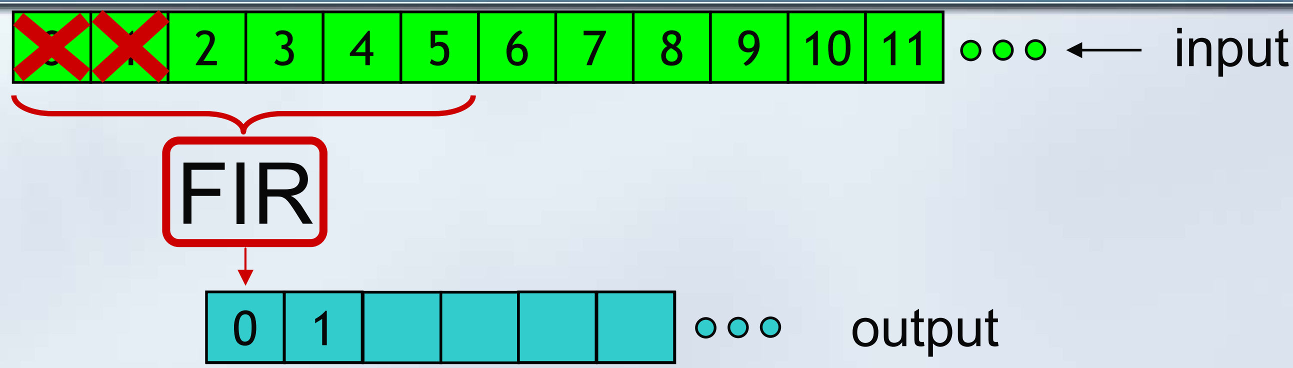


4. StreamIt - Filter

- Osnovna računska jedinica
- Svaki filter posjeduje work funkciju
- Svaki filter ima ulaznu traku i izlaznu traku
- Svaki filter definira brzinu pristupa ulaznoj i izlaznoj traci (pop, push i peek rate)

```
float→float filter sum2elems () {
        work pop 2 push 1() {
                int first = pop();
                int second = pop();
                push(first + second);
        }
}
```

4. StreamIt filter (FIR primjer)




```
float→float filter FIR  (int N, float[N] weights) {
    work push 1 pop 1 peek N {
        float result = 0;

        for (int i = 0; i < N; i++) {
            result += weights[i] * peek(i);
        }
        pop();
        push(result);
    }
}
```

4. StreamIt (FIR primjer u C-u)

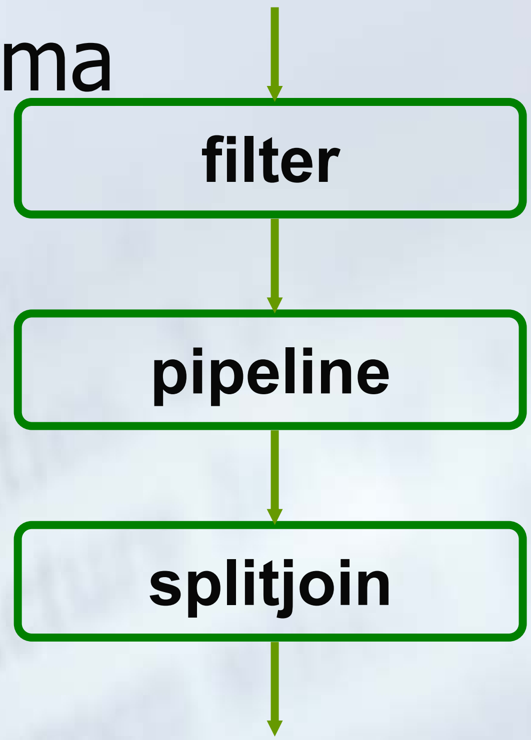
```
void FIR(
  int* src,
  int* dest,
  int* srcIndex,
  int* destIndex,
  int srcBufferSize,
  int destBufferSize,
  int N) {

  float result = 0.0;
  for (int i = 0; i < N; i++) {
    result += weights[i] * src[(*srcIndex + i) % srcBufferSize];
  }
  dest[*destIndex] = result;
  *srcIndex = (*srcIndex + 1) % srcBufferSize;
  *destIndex = (*destIndex + 1) % destBufferSize;
}
```

Čitanje podataka (buffer management) izmješano sa funkcionalnošću

4. StreamIt - pipeline

- Slijedni niz komponenti
- Izlaz jedne komponente spojen na ulaz sljedeće u nizu
- Izražavanje protočnog paralelizma

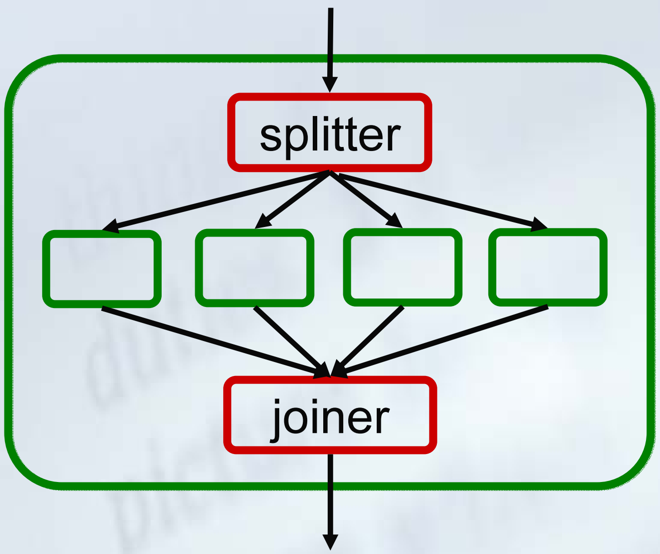
```
float→float pipeline 2D_iDCT (int N)
{
        add Column_iDCTs(N);
        add Row_iDCTs(N);
}
```



4. StreamIt – splitjoin

- Scatter-gather struktura
- Izražavanje DLP i TLP paralelizma
- Postoji više načina razvrstavanja i skupljanja podataka (duplicate, roundrobin)

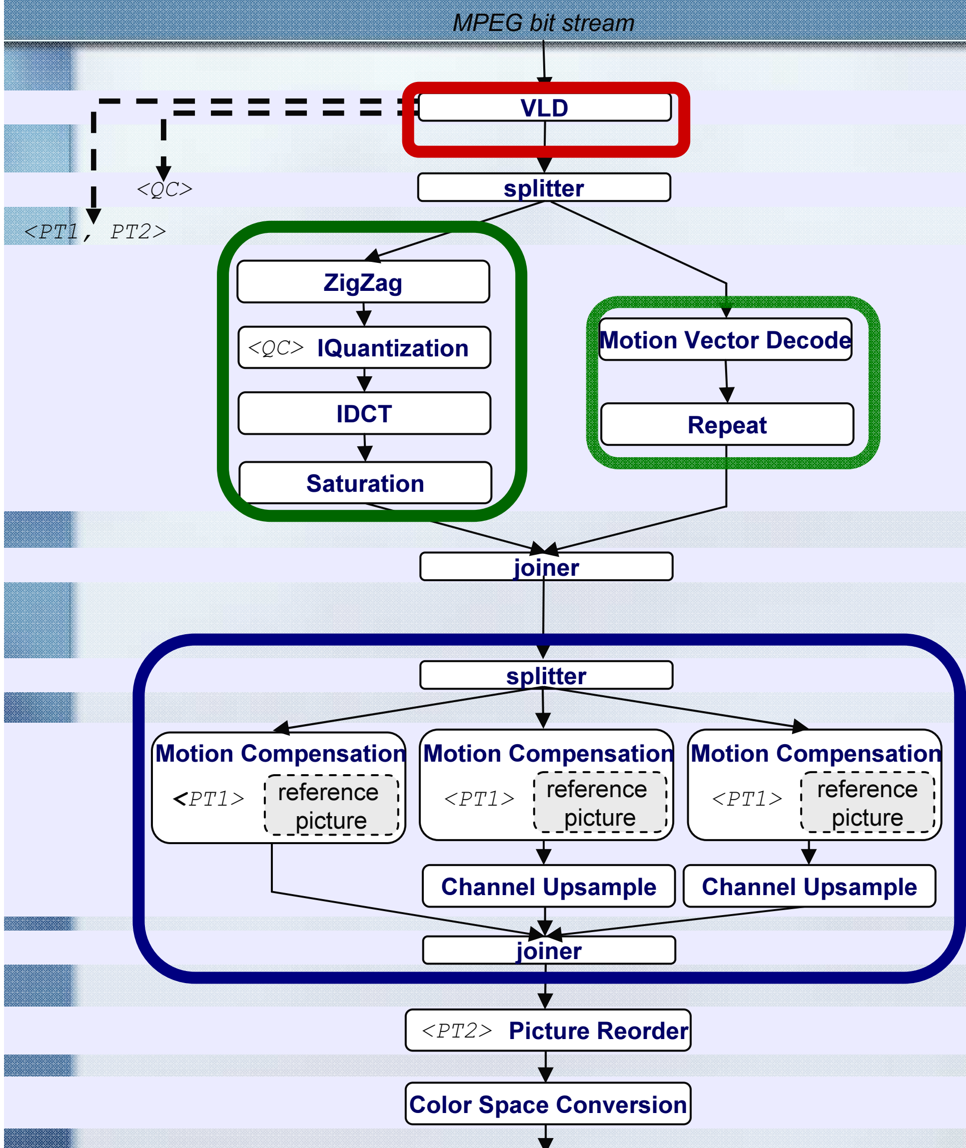
```
float→float splitjoin Row_iDCT (int N)
{
    split roundrobin(N);
    for (int i = 0; i < N; i++) {
        add 1D_iDCT(N);
    }
    join roundrobin(N);
}
```



4. StreamIt – MM aplikacije

- StreamIt je prikladan za aplikacije:
  - Sa velikom količinom podataka (tok podataka)
  - Izvođenje grafa toka je stabilno u vremenu, tj. graf toka se ne mijenja prečesto (podatkovni tok dominira nad kontrolnim tokom)
  - Nezavisne operacije: graf toka je po prirodi paralelan, i program se može prikazati kao skup nezavisnih operacija nad podatcima (filteri)
  - DSP, kodiranje slike i videa, mrežne obrade, kriptografija, grafička obrada...

4. MPEG-2 StreamIt



filters

pipelines

splitjoins

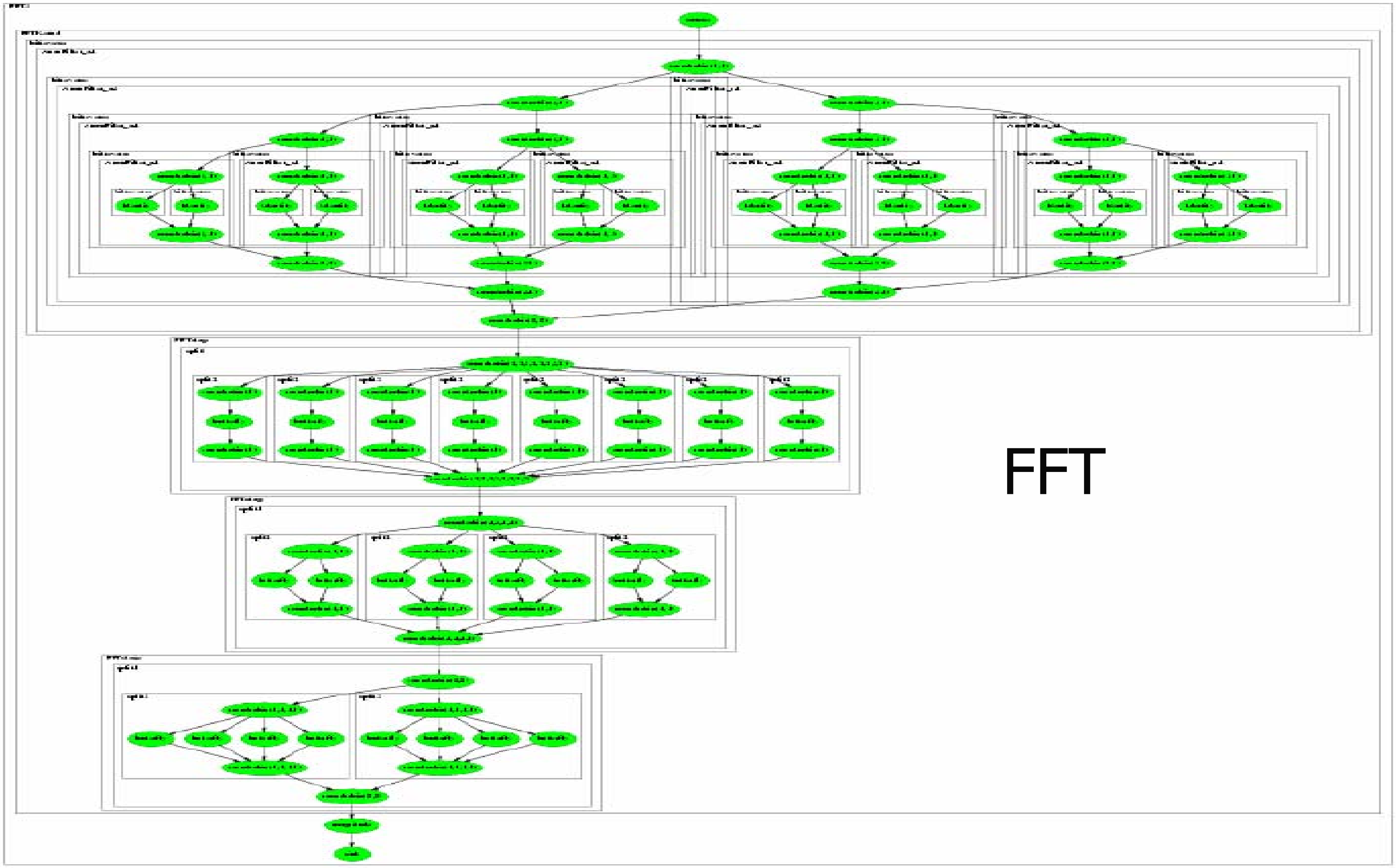
```
add VLD(QC, PT1, PT2);
add splitjoin {
    split roundrobin(N*B, V);

    add pipeline {
        add ZigZag(B);
        add IQuantization(B) to QC;
        add IDCT(B);
        add Saturation(B);
    }
    add pipeline {
        add MotionVectorDecode();
        add Repeat(V, N);
    }

    join roundrobin(B, V);
}
add splitjoin {
    split roundrobin(4*(B+V), B+V, B+V);

    add MotionCompensation(4*(B+V)) to PT1;
    for (int i = 0; i < 2; i++) {
        add pipeline {
            add MotionCompensation(B+V) to PT1;
            add ChannelUpsample(B);
        }
    }

    join roundrobin(1, 1, 1);
}
add PictureReorder(3*W*H) to PT2;

add ColorSpaceConversion(3*W*H);
```

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

4. StreamIt - Zaključak

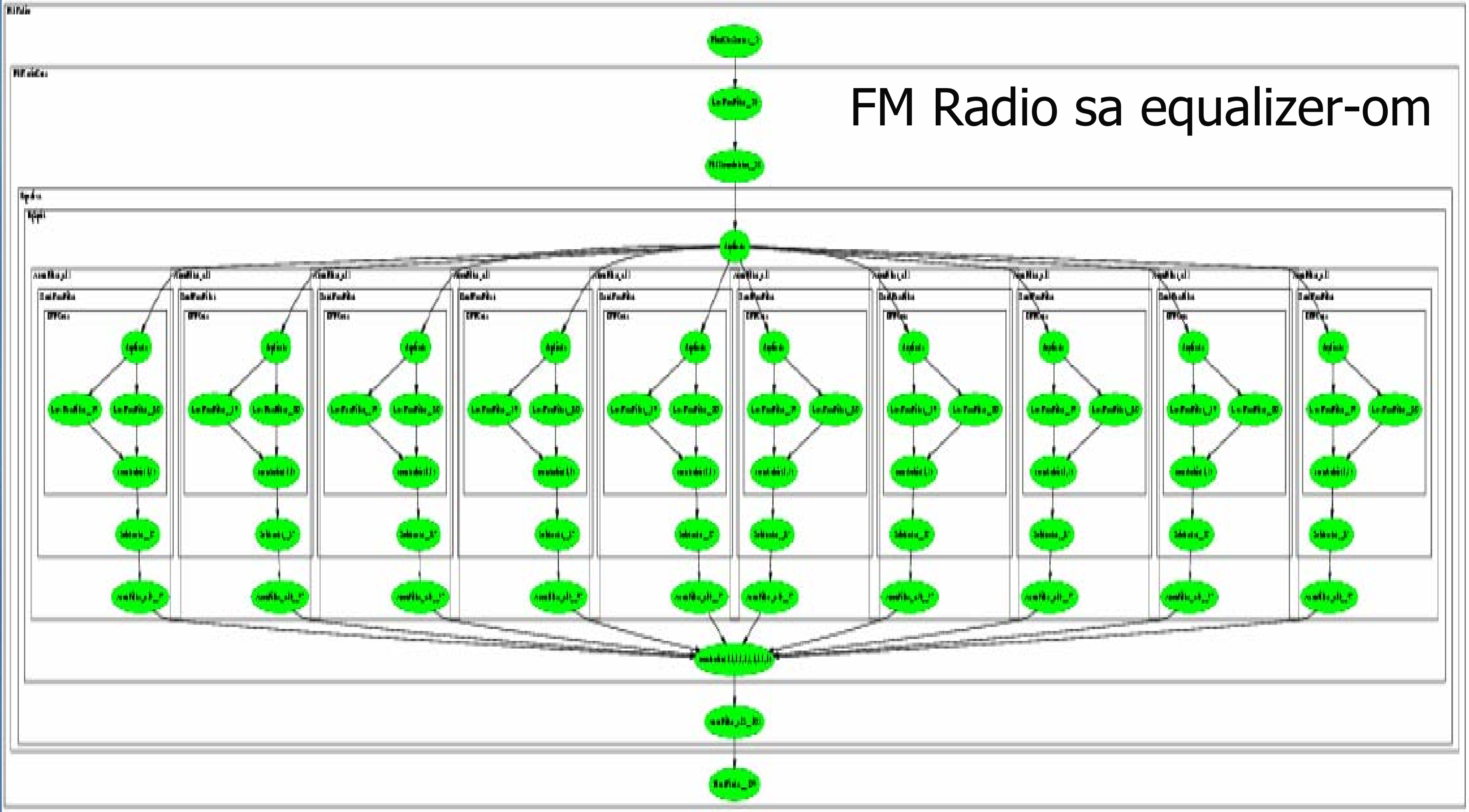
- Program prikazan kao graf nezavisnih komponenti (filter, splitjoin, pipeline, feedback)
- Komunikacija među komponentama: eksplicitna
- Paralelizam: eksplicitan iz grafa toka (stream graph)
- Prevodioc: omogućen agresivan pristup optimizaciji grafa toka (stapanje filtera, razbijanje filtera, scatter-gather itd.) prilagođenog ciljanoj arhitekturi
- Prirodan način izražavanja paralelizma u MM aplikacijama

4. StreamIt – Zaključak



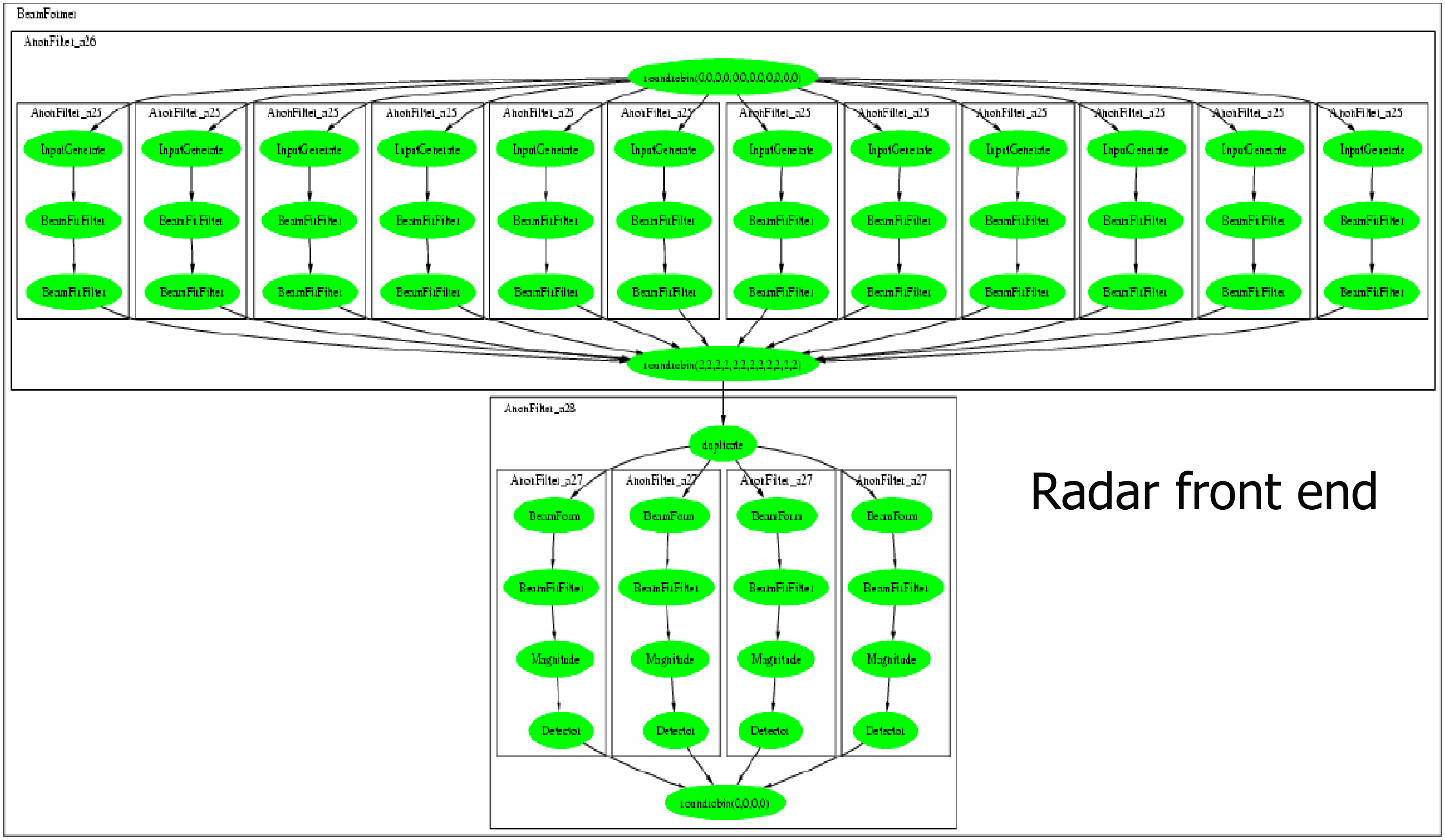
FFT

4. StreamIt – Zaključak



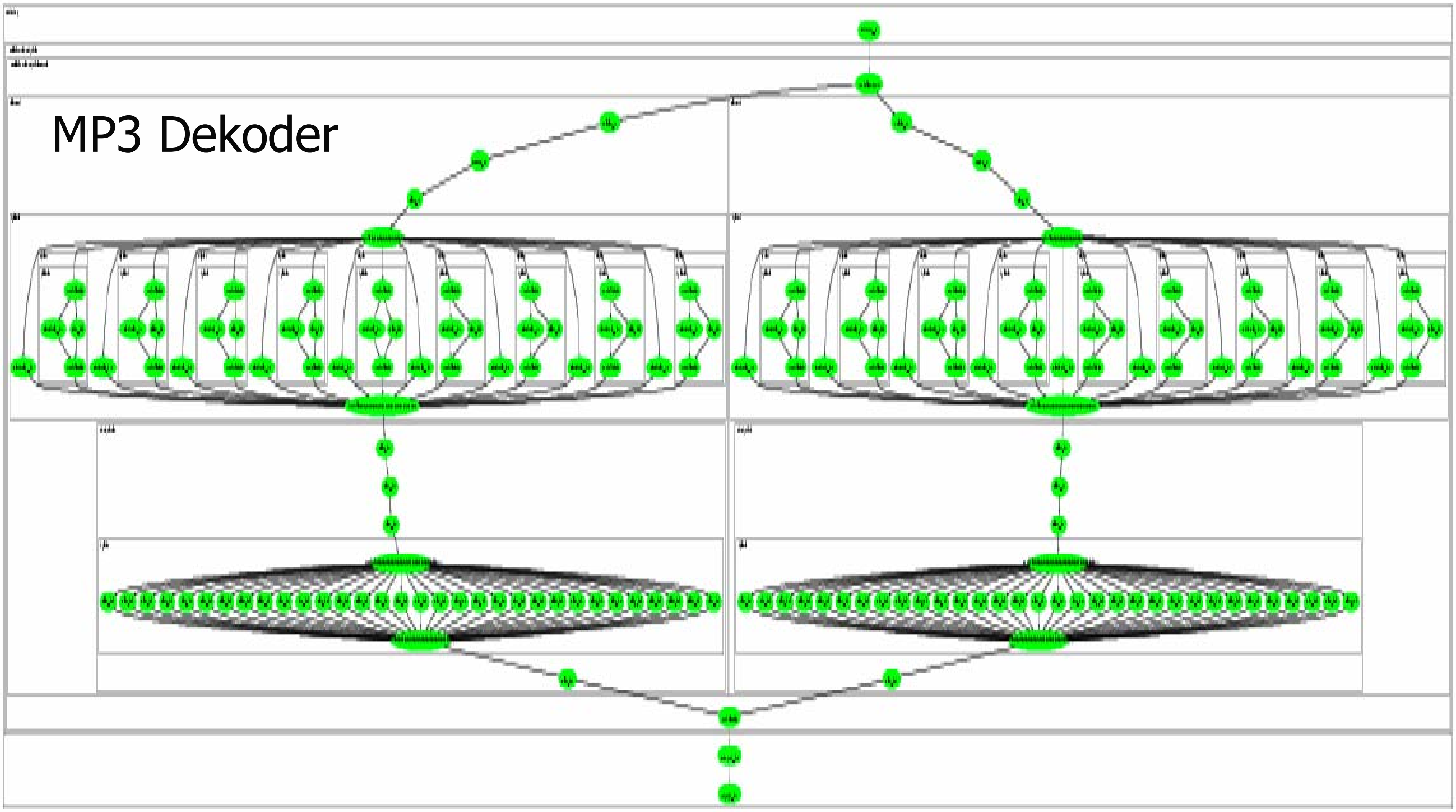
FM Radio sa equalizer-om

4. StreamIt – Zaključak



Radar front end

4. StreamIt – Zaključak



4. StreamIt - Zaključak

- Za one koji žele znati više:
  - StreamIt project: CSAIL MIT: http://cag.csail.mit.edu/streamit
- Slični projekti:
  - BrookGPU: Stanford: http://graphics.stanford.edu/projects/brookgpu/
  - Ptolemy: Berkeley: http://ptolemy.eecs.berkeley.edu/
  - Cg: NVIDIA: http://developer.nvidia.com/page/cg_main.html
  - StreamC/KernelC
  - Lustre i Esterel
  - Spindle

ZAKLJUČAK