



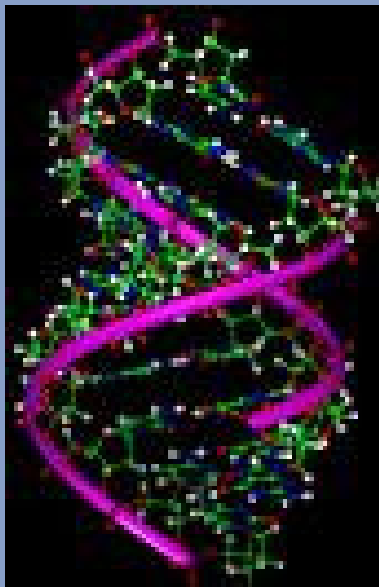
Universiteit Leiden  
The Netherlands

# BA Projects in the CS Group (LERC)

Dr ir Bart Kienhuis  
Assistant Professor / Leiden University, LIACS  
([www.liacs.nl/~kienhuis](http://www.liacs.nl/~kienhuis))

Universiteit Leiden. University to discover.

# Stream Based Applications



Bio informatics



Multi-Media  
(TV on PDAs)

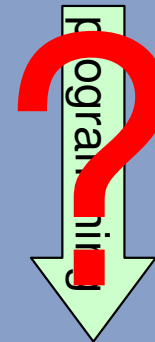


Classical DSP  
(SDR/Beam Former)



Imaging  
(Smart Cameras)

Applications  
(C / Matlab / Java)



Multi  
Core  
System  
(FPGA)

***We have these fantastic new applications and architectures; we just need to program them***

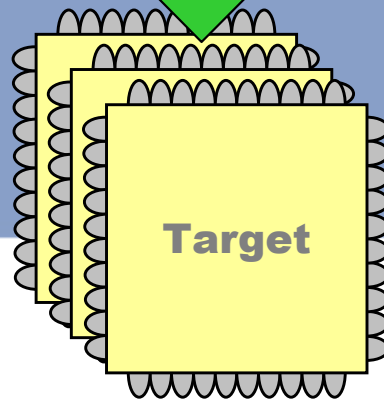
# Our look on the World

Application

Given: Tomograph  
BioComputing, DSP

*CS Group:* tools, methods  
to map given applications  
onto given multicore  
architectures

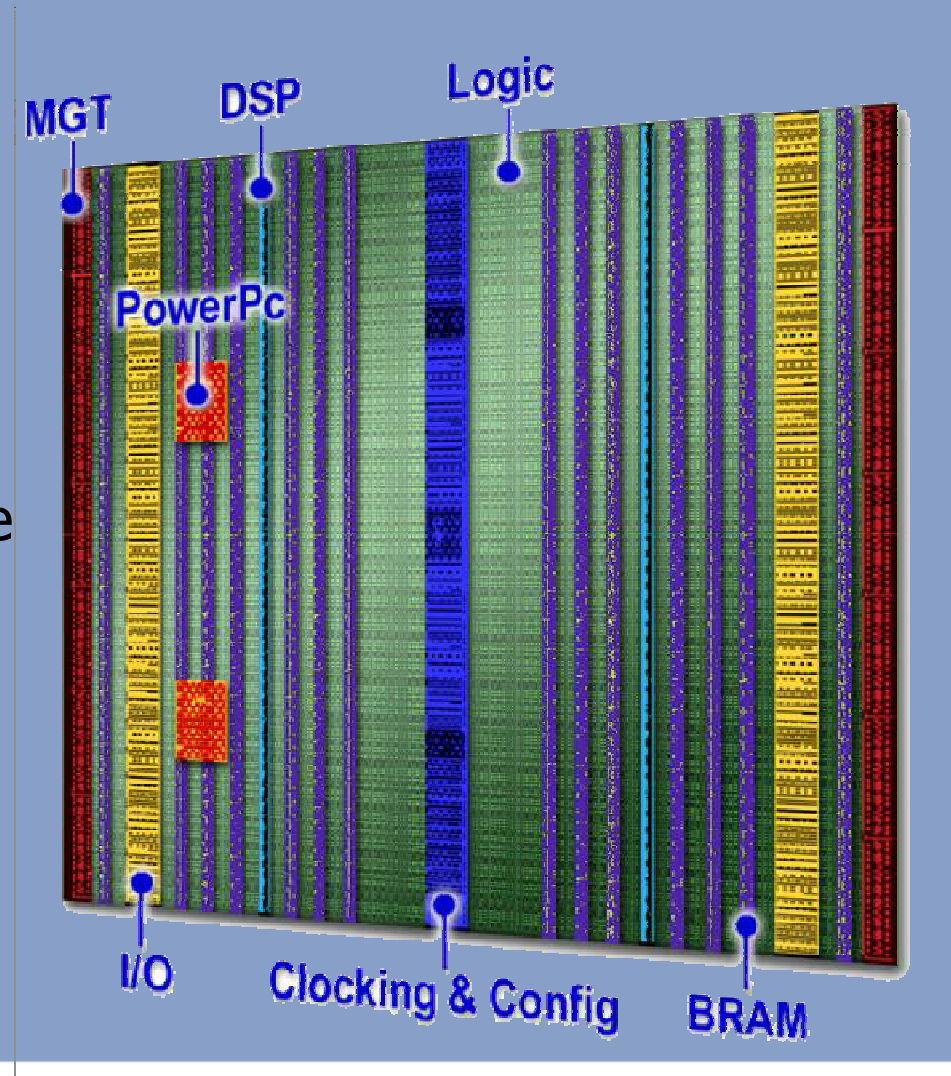
Mapping of stream based  
Applications on  
multi processor  
Systems



Given: Cell Processor  
FPGAs, NVidia GPU

# FPGA Technology

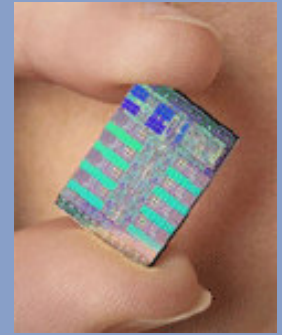
- FPGAs already have over a billion transistors!
- Combine various components
  - Programmable Hardware
  - CPUs (Software)
  - Distributed Memory
- A truly heterogeneous multi core platform



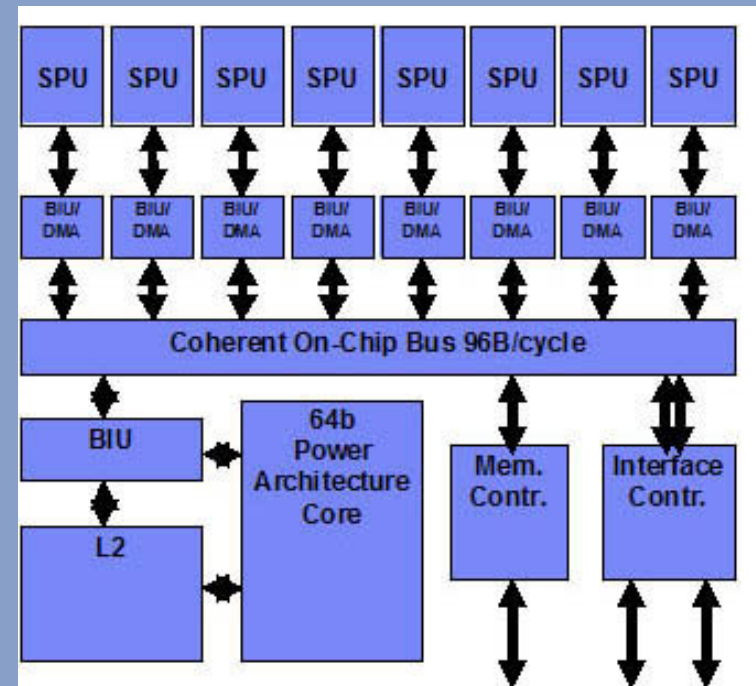
ALTERA

XILINX

# Cell Processor



- Cell processor -> playstation 3
- 9 processors, 1 PowerPC (PPU), 8 dedicated SPEs
- Open Source support by IBM



# Compaan Technology

## Matlab / C / Java

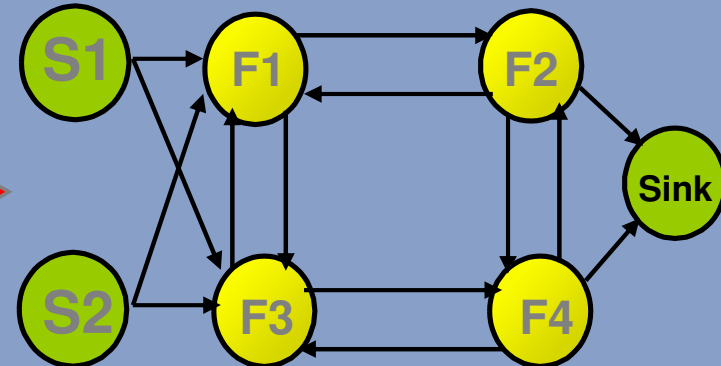
```
for j = 1:1:N,  
    [x(j)] = S1( );  
end  
for i = 1:1:K,  
    [y(i)] = S2( );  
end  
for j = 1:1:N,  
    for i = 1:1:K,  
        [y(i), x(j)] = func(y(i), x(j) );  
    end  
end  
for i = 1:1:K,  
    [Out(i)] = Sink( y( I ) );  
end
```

*Parameterized Nested  
Loop Programs*

**Compaan**

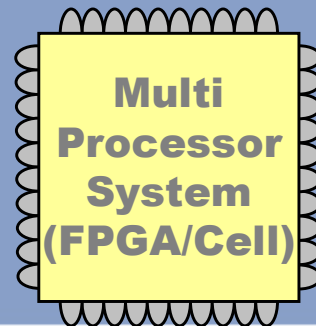


## Kahn Process Network (KPN)



*Pipelined Stream-based  
Architecture*

*Distributed memory  
and control*



**Cellcc  
Espam**

# BA Projects

## Matlab / C / Java

```
for j = 1:1:N,  
    [x(j)] = S1( );  
end  
for i = 1:1:K,  
    [y(i)] = S2( );  
end  
for j = 1:1:N,  
    for i = 1:1:K,  
        [y(i), x(j)] = func(y(i), x(j) );  
    end  
end  
for i = 1:1:K,  
    [Out(i)] = Sink( y( I ) );  
end
```

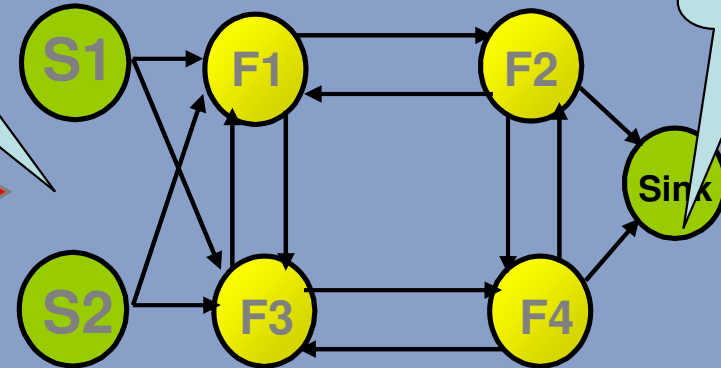
IDE support in Eclipse

Compaan Compiler development

Compaan



## Kahn Process Network (KPN)



Modeling in SystemC

Espam

Cellcc

Integration in Eclipse

Hardware extensions

Case Studies

Multi Processor System (FPGA)

Cell Case Study

Multi Processor System (Cell)

# Compaan Technology

## Matlab / C / Java

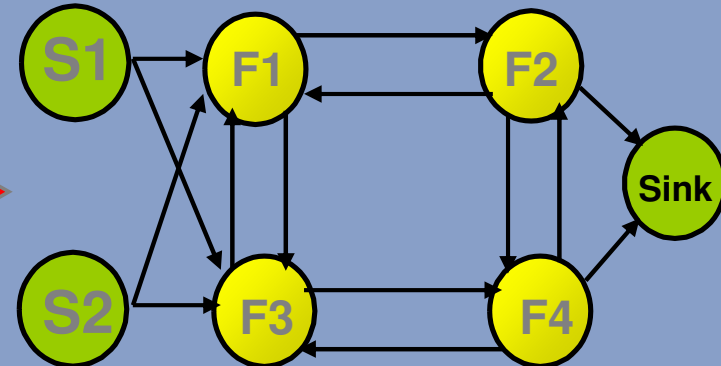
```
for j = 1:1:N,  
    [x(j)] = S1( );  
end  
for i = 1:1:K,  
    [y(i)] = S2( );  
end  
for j = 1:1:N,  
    for i = 1:1:K,  
        [y(i), x(j)] = func(y(i), x(j) );  
    end  
end  
for i = 1:1:K,  
    [Out(i)] = Sink( y( I ) );  
end
```

*Parameterized Nested  
Loop Programs*

**Compaan**

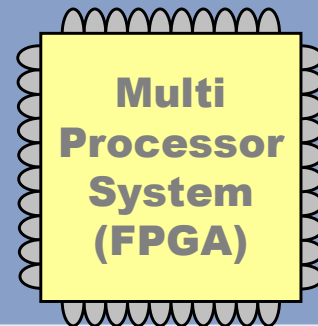


## Kahn Process Network (KPN)



*Pipelined Stream-based  
Architecture*

*Distributed memory  
and control*



**Laura  
Espam**

*Compaan is the technology developed at Leiden University  
to tackle the Programming Productivity Gap for FPGAs*

# BA Projects (Last Year)

- Integrating the 'Compaan' compiler as an Eclipse Plug-in
- Creating a System-C back-end for Compaan
- Creating a Kahn Process Network in the dynamic dataflow formalism
- Polytope visualizatie mbv Open GL en SWT
- Multi-core simulatie
- Code generatie met Eclipse JET framework
- Applicatie afbeelden op PS3
- Schrijf een Eclipse highlight editor

<http://www.liacs.nl/home/kienhuis/bach/projects.html>

# CS Groep (LERC)

- Prof. Ed Deprettere
- Dr. Bart Kienhuis
- Dr. Todor Stefanov
- Hristo Nikolov
- Sjoerd Meijer
- Jerome Lemaitre
- Bin Jiang
- Dmitry Nadezhkin

