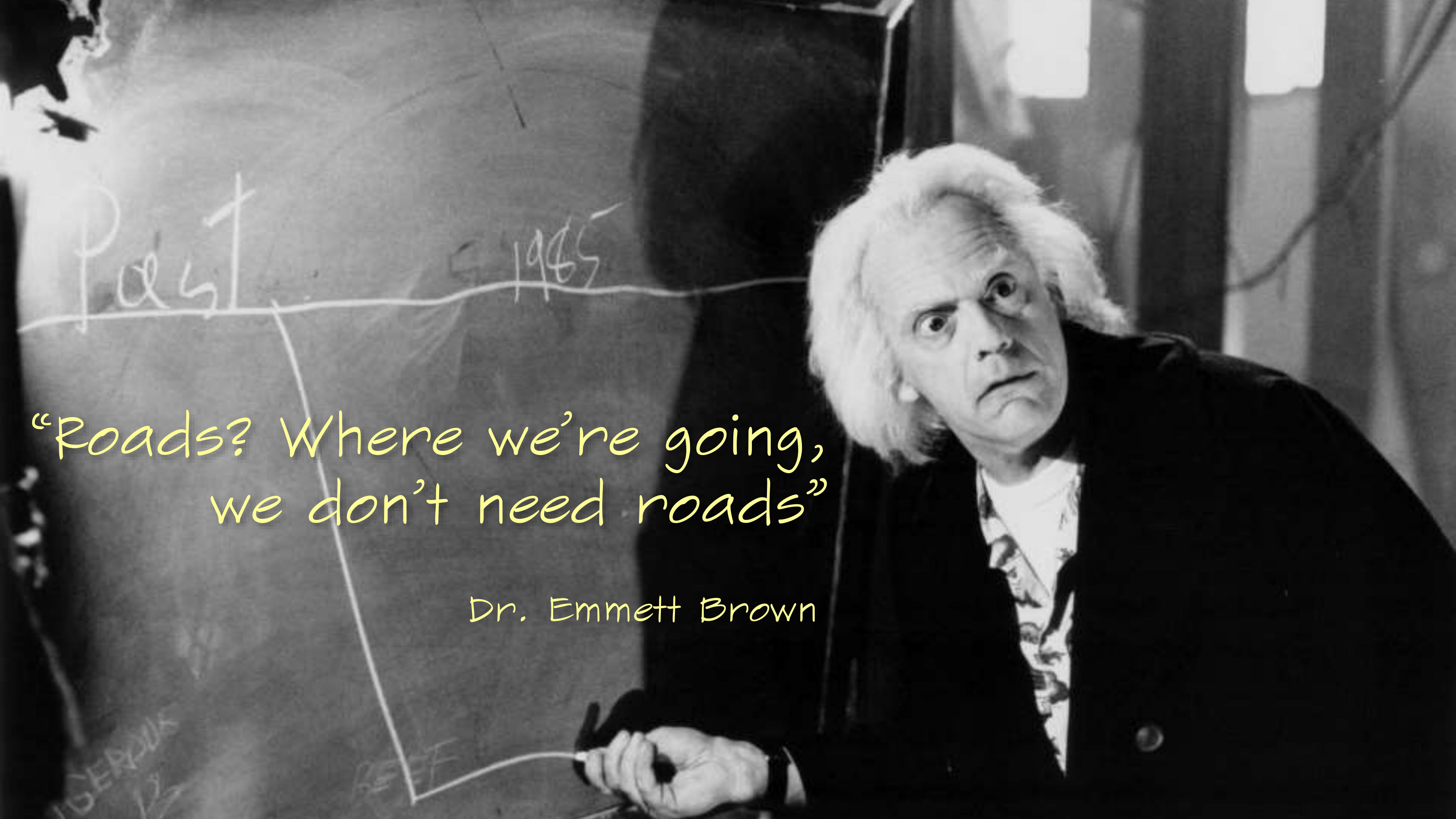
HPEC
(High Performance Embedded Computing)

Overview

Bordeaux INP
ENSEIRB
MATMECA

J. CRENNE – 2020/2021


Past
1985
“Roads? Where we’re going, we don’t need roads”
Dr. Emmett Brown

But, meanwhile we have a deadly schedule

What you'll have to do !

6 Labs x 4H00

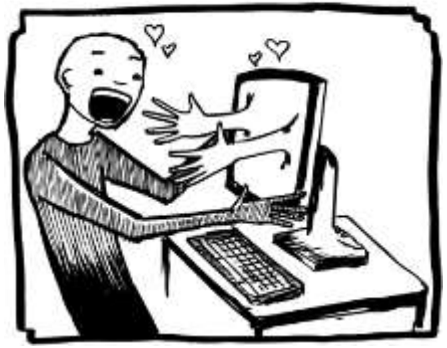
Lab/Project	Lab/Project	Lab/Project	Lab/Project	Lab/Project	Lab/Project

- ❑ 6 (or 7, or even 8 !) Labs supervised by your favorite teacher
- ❑ You'll work in pair
- ❑ No daily report
- ❑ Last Lab :
  - Final report (~10-15 pages) + source code
  - Demo/presentation with final report submission (~15 min)

Goal of this course

What you'll have to do !

Implement a math intensive application on an embedded System-on-Chip (SoC). But not only…

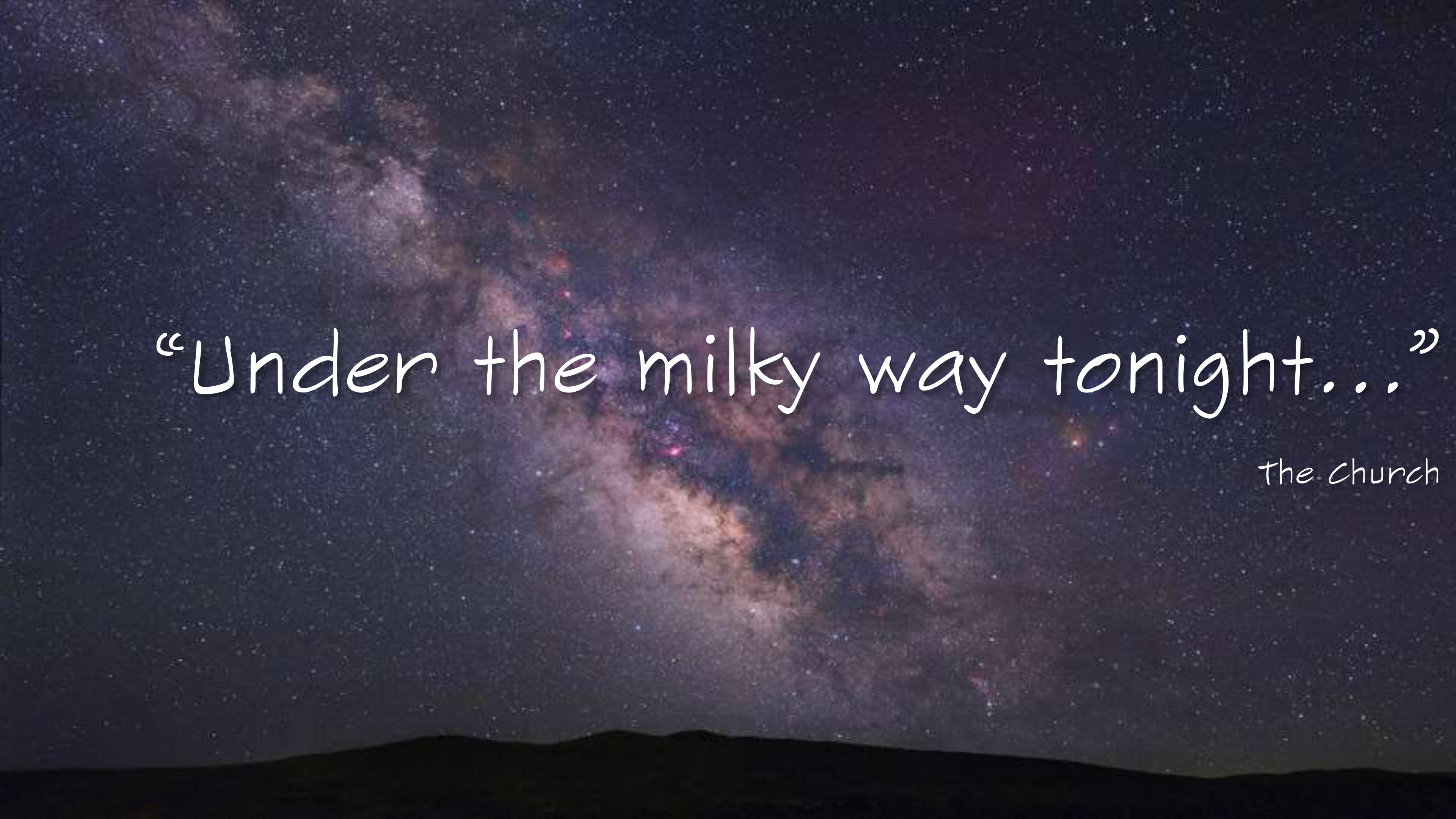


This application will be visual and interactive !


GALAXEIRB

High-level Galaxies Simulation Modeling

http://en.wikipedia.org/wiki/N-body_simulation
http://www.kof.zcu.cz/st/dis/schwarzmeier/galaxy_interactions.html


"Under the milky way tonight..."
The Church

High-level Galaxies Simulation Modeling

Overview

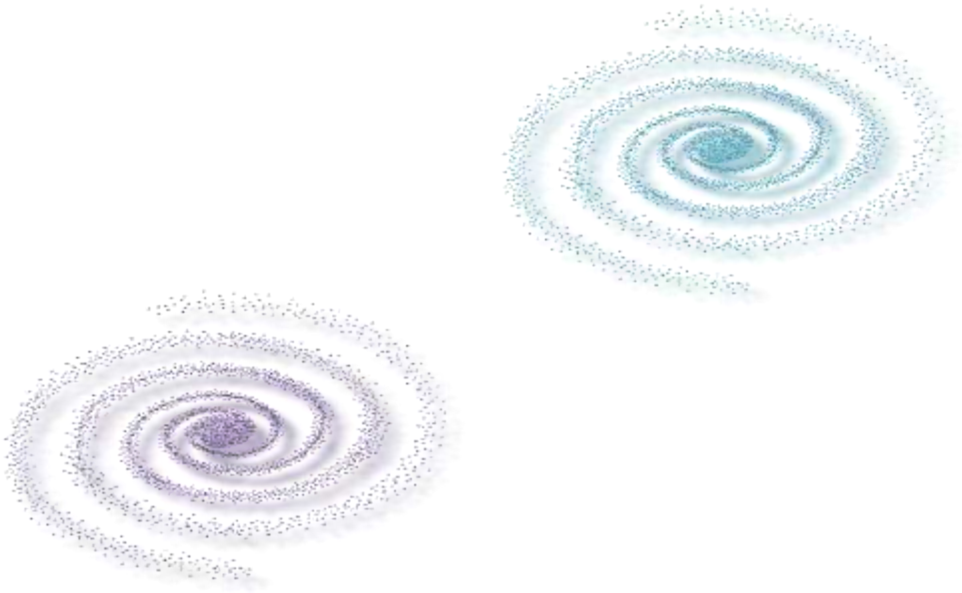


Figure. Two galaxies and their gravitational influences

High-level Galaxies Simulation Modeling

Overview

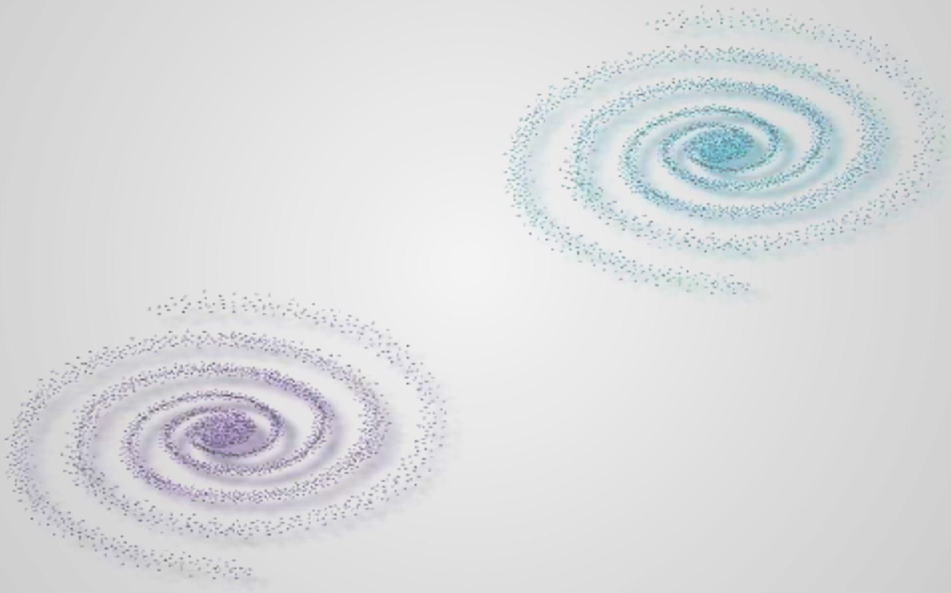


Figure. Two galaxies and their gravitational influences

High-level Galaxies Simulation Modeling

Overview

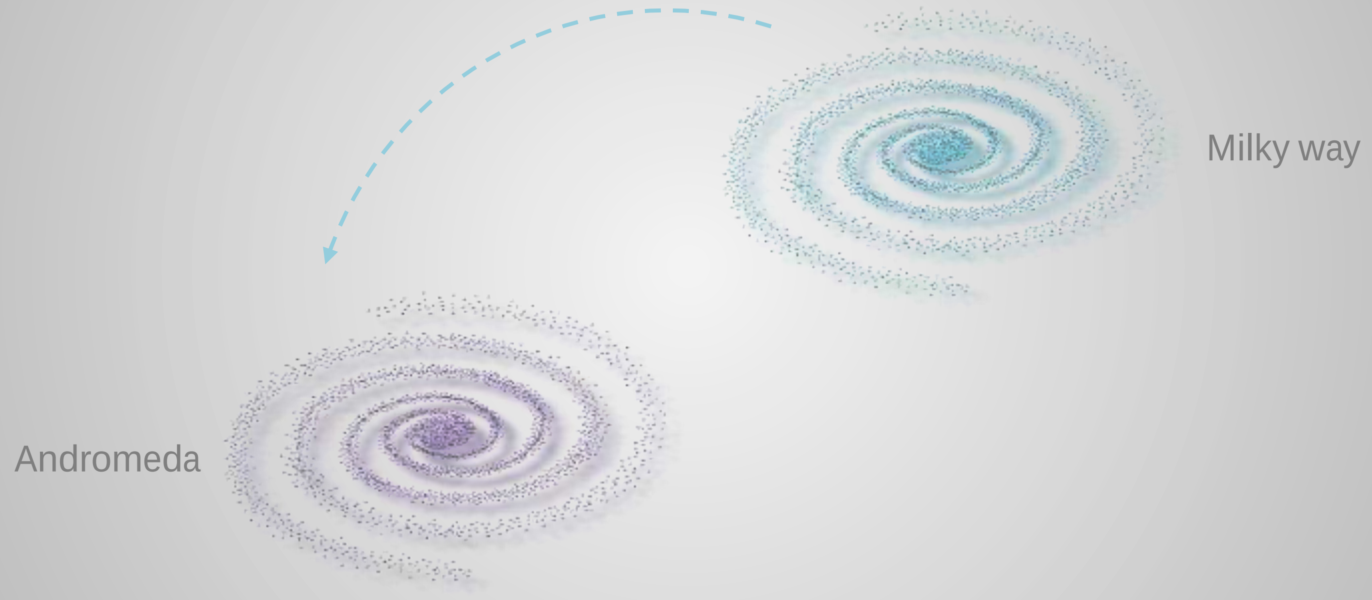


Figure. Two galaxies and their gravitational influences

High-level Galaxies Simulation Modeling

Overview

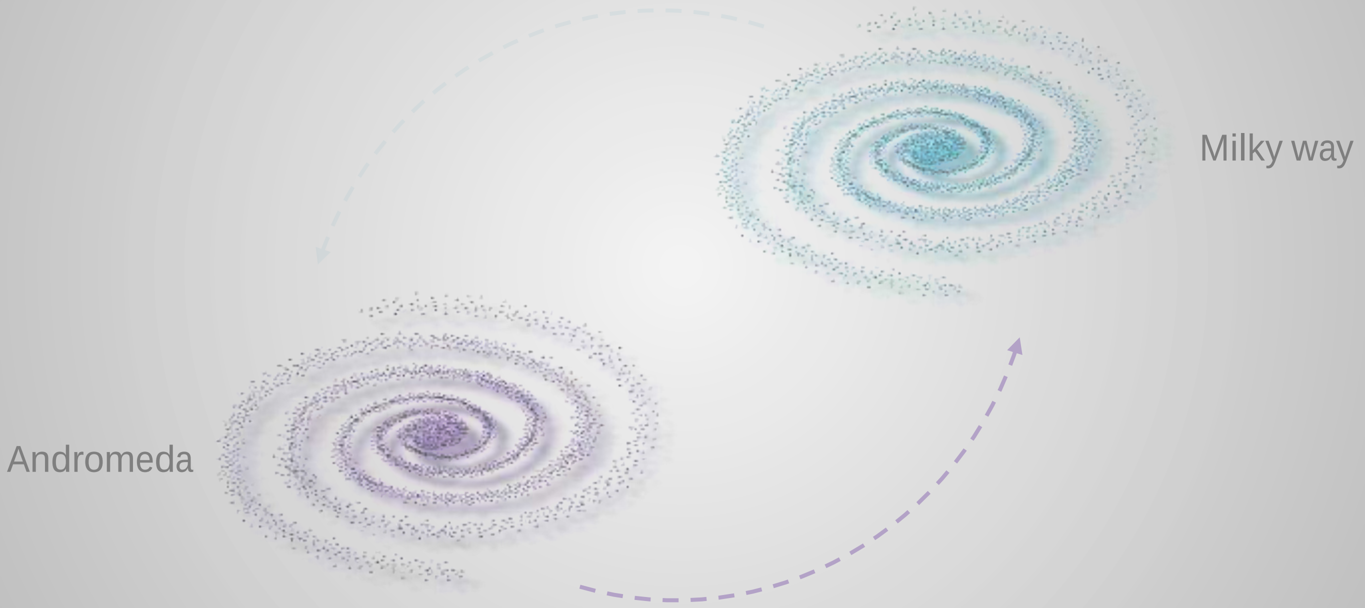


Figure. Two galaxies and their gravitational influences

High-level Galaxies Simulation Modeling

Spiral galaxies morphology

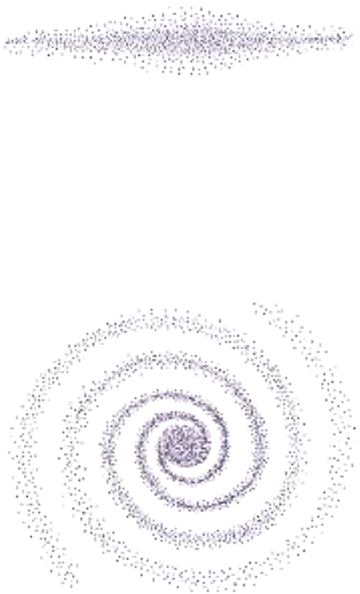


Figure. Face and side views of a disk galaxy

High-level Galaxies Simulation Modeling

Spiral galaxies morphology

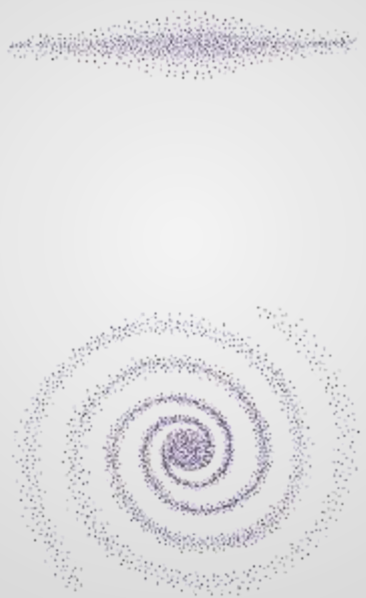


Figure. Face and side views of a disk galaxy

High-level Galaxies Simulation Modeling

Spiral galaxies morphology

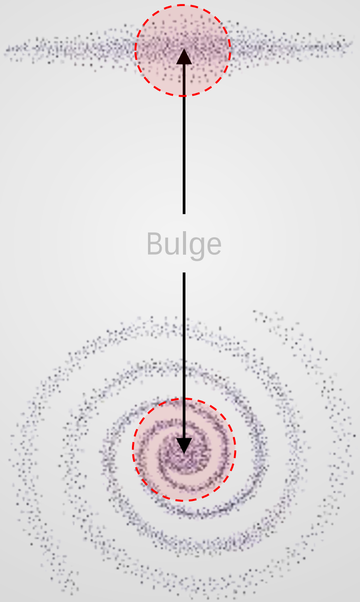


Figure. Face and side views of a disk galaxy

High-level Galaxies Simulation Modeling

Spiral galaxies morphology

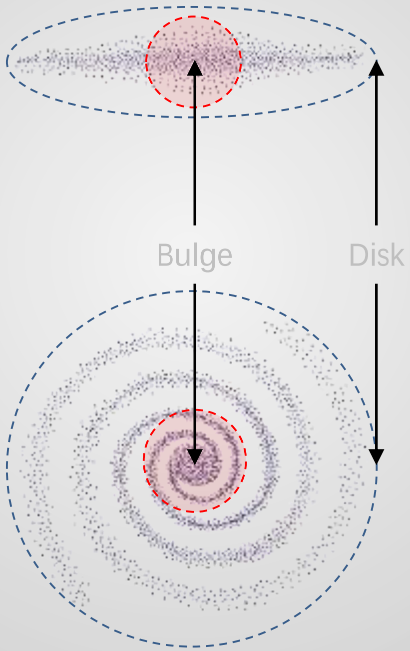


Figure. Face and side views of a disk galaxy

High-level Galaxies Simulation Modeling

Spiral galaxies morphology

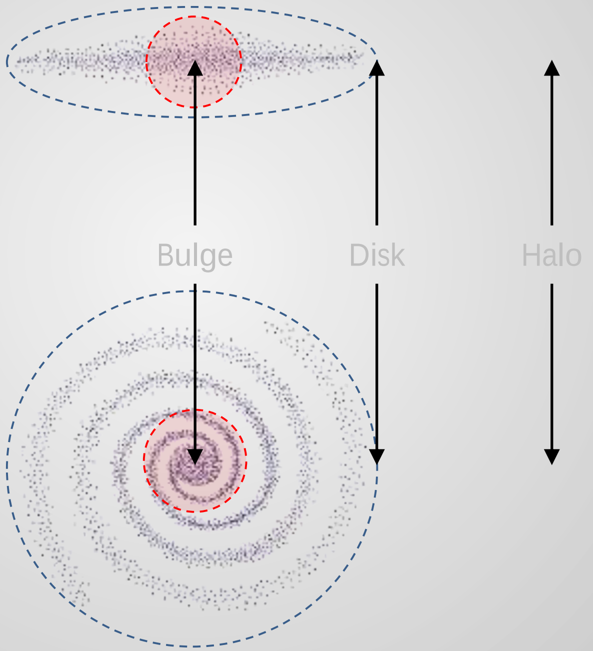


Figure. Face and side views of a disk galaxy

Particle Modeling

Particle Modeling

Characterization and coordinate system

Each celestial body is a particle characterized by its position, velocity and mass. To represent a particle in a three-dimensional (3D) space, a Cartesian coordinate system is conventionally used

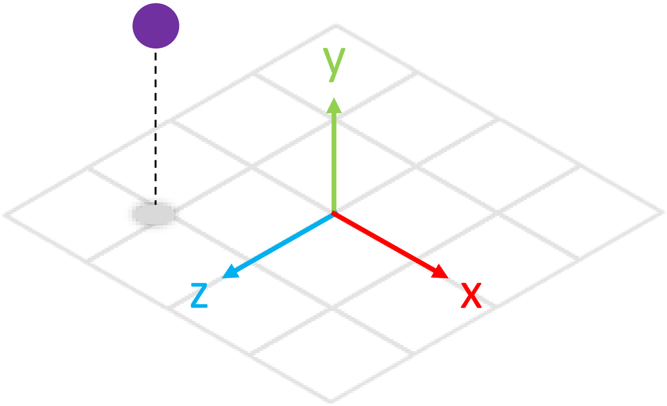


Figure. A three-dimensional space using a cartesian coordinate system, AKA « right-handed » system

Particle Modeling

Motion

The motion of a particle i is done in a Cartesian coordinates system

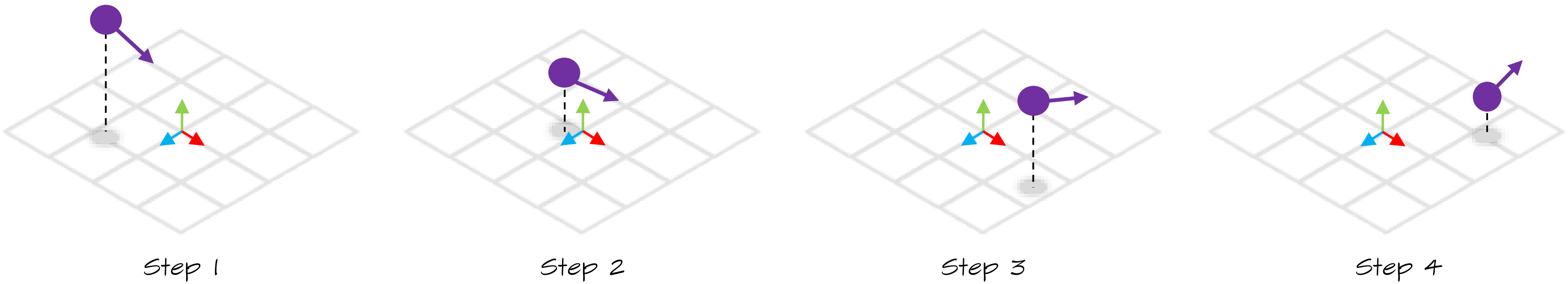


Figure. A particle is moving in a Cartesian system

This motion is not only influenced by its own mass and velocity but also by positions and masses of other particles in the given "bodies" set S

The position of a particle at time $t+1$ is then computed by summing every particle contributions in set S. It is a "brute force" method.

Although very accurate, this simulation model exhibits a time complexity of $O(n^2)$

Particle Modeling

Contribution and complexity

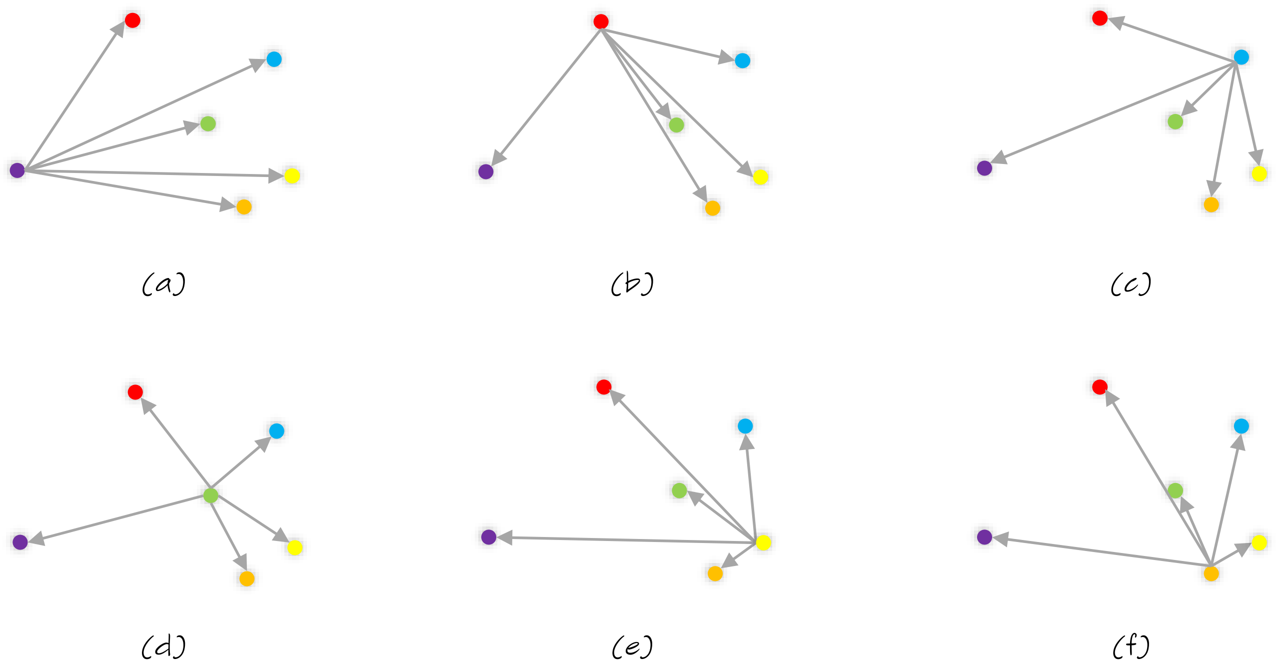


Figure. Five particles in purple, red, blue, green, yellow and orange

Particle Modeling

Contribution and complexity

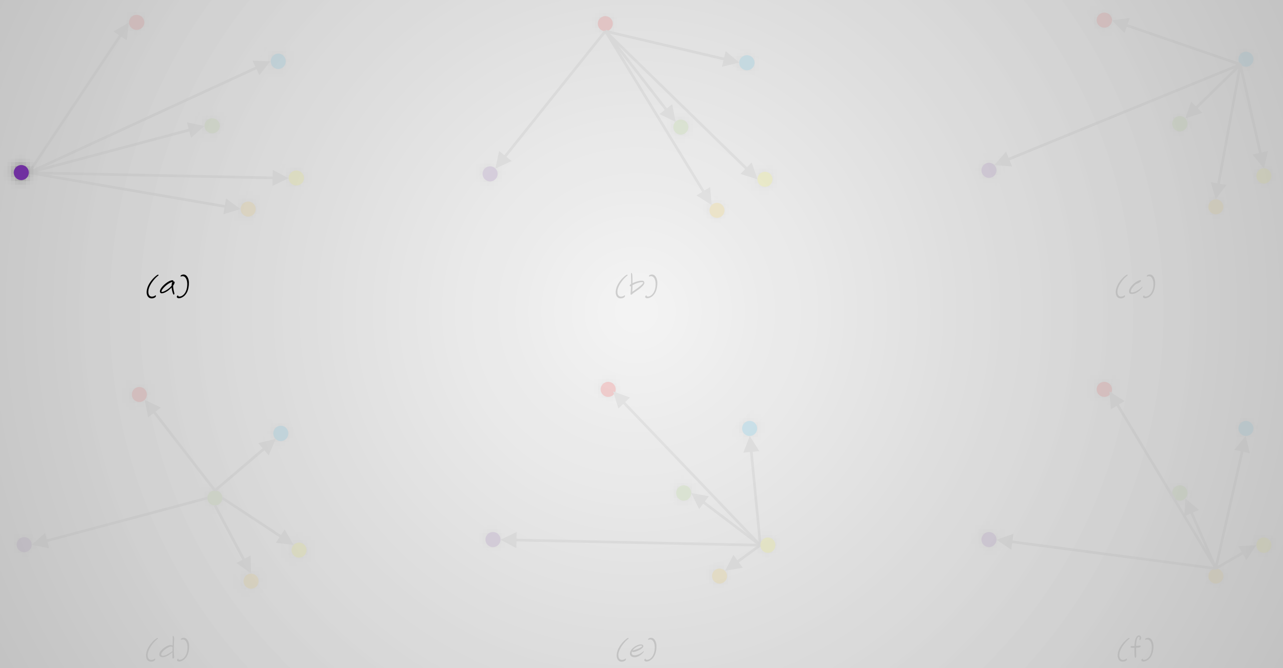


Figure. Five particles in purple, red, blue, green, yellow and orange

Particle Modeling

Contribution and complexity

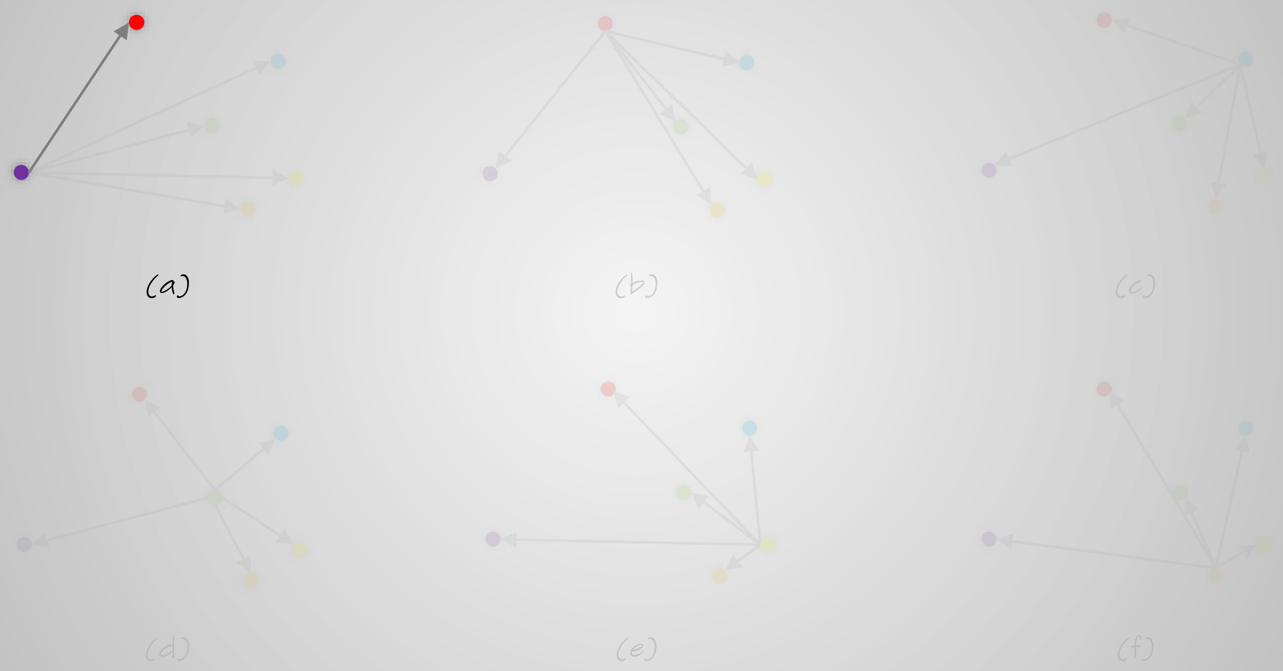


Figure. Five particles in purple, red, blue, green, yellow and orange

Particle Modeling

Contribution and complexity

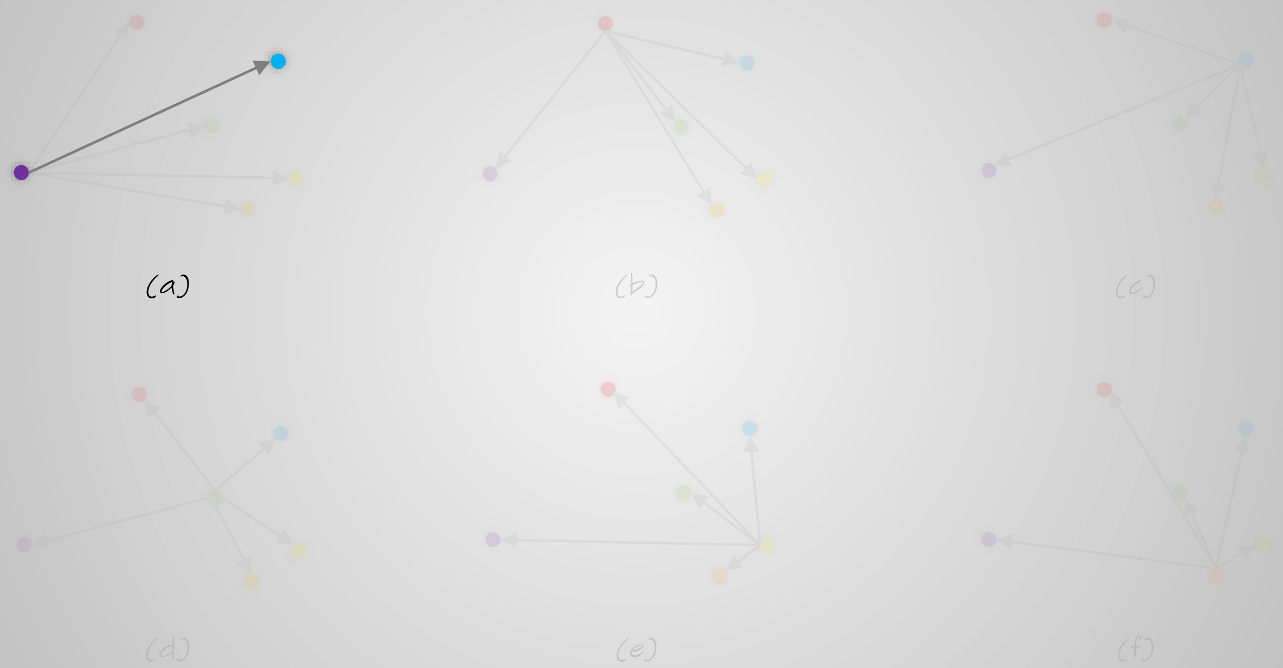


Figure. Five particles in purple, red, blue, green, yellow and orange

Particle Modeling

Contribution and complexity

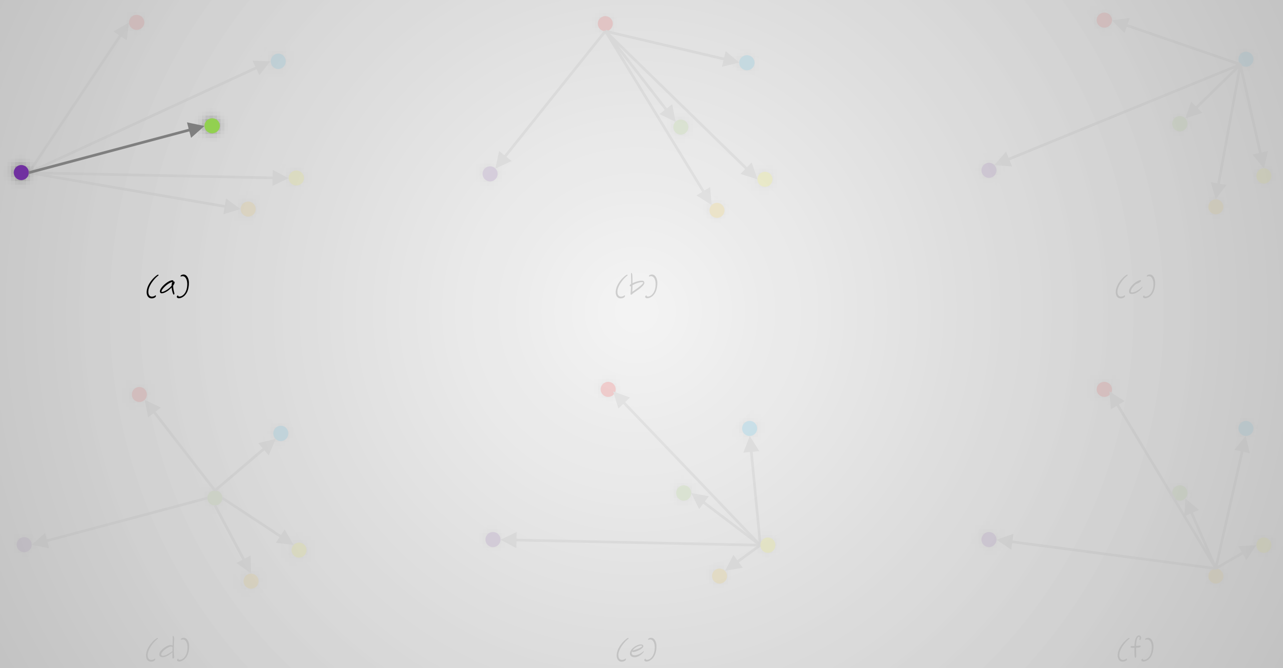


Figure. Five particles in purple, red, blue, green, yellow and orange

Particle Modeling

Contribution and complexity

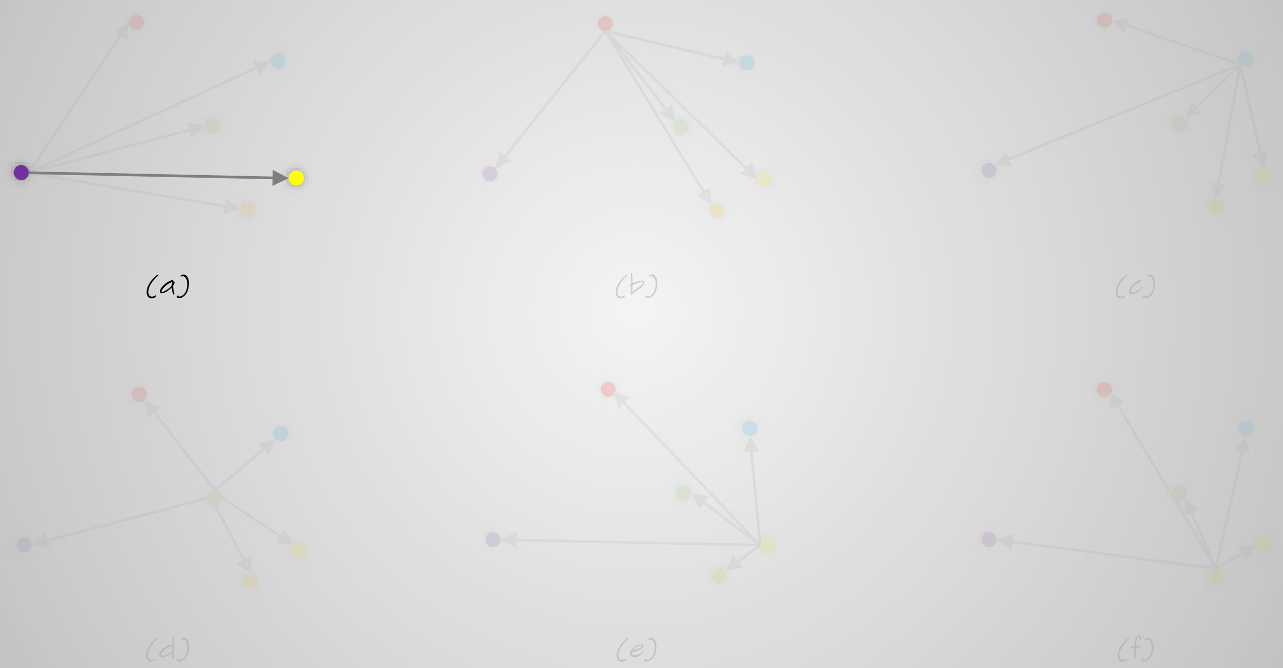


Figure. Five particles in purple, red, blue, green, yellow and orange

Particle Modeling

Contribution and complexity

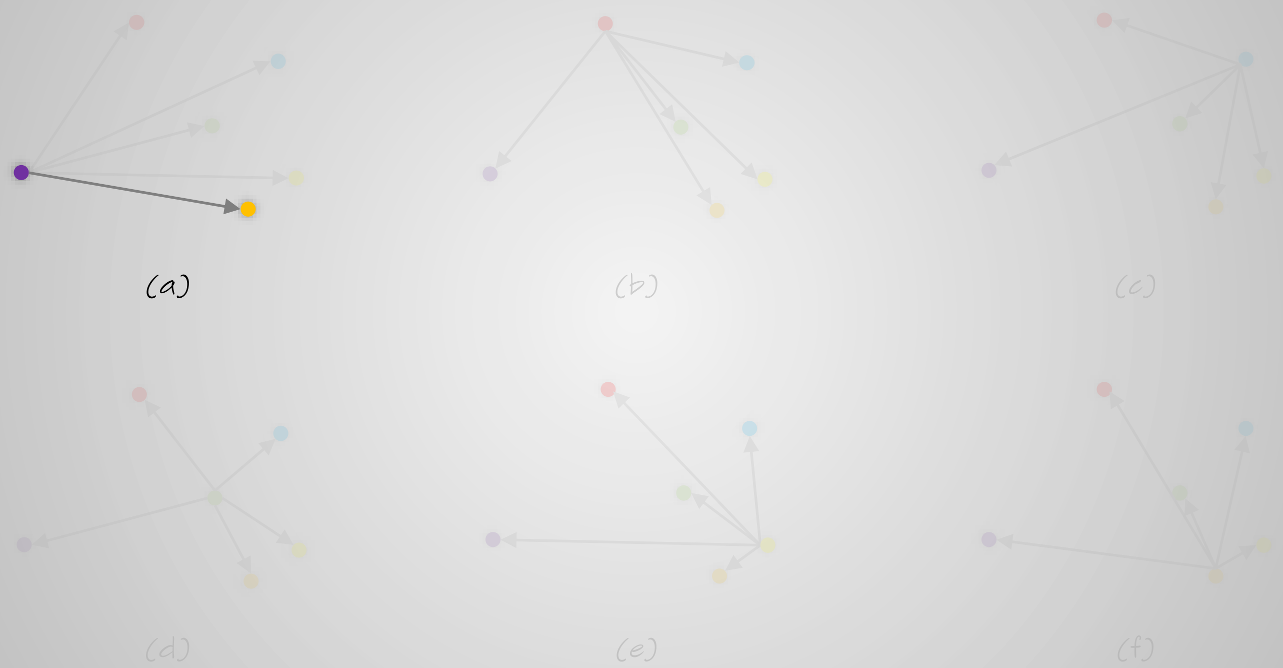


Figure. Five particles in purple, red, blue, green, yellow and orange

Particle Modeling

Damping factor

The damping factor ζ influences a particle velocity over time.

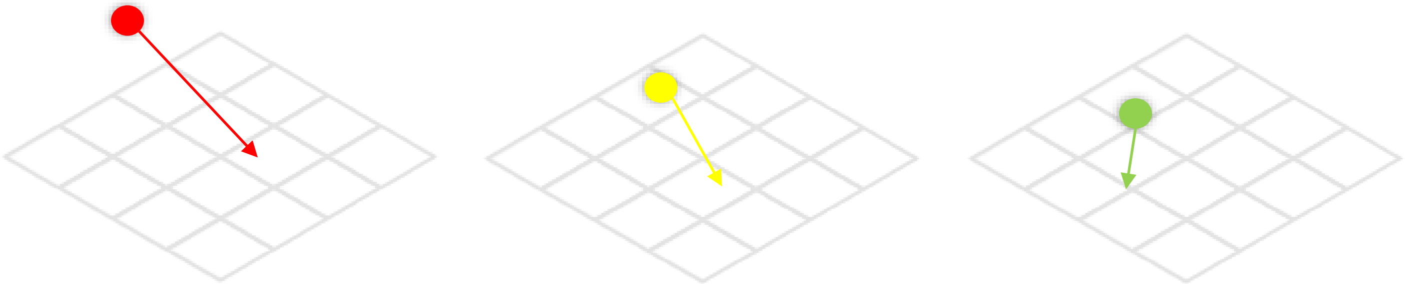


Figure. Step by step influence of the damping factor ζ on one particle

A ζ factor < 1.0 decreases a particle velocity while a ζ factor > 1.0 increases a particle velocity.

Particle Modeling

Simulation algorithm

```
FOR each step DO
  FOR each particle DO
    particle acceleration ← 0
    FOR each neighbor particle DO
      particle acceleration ← Add Acceleration ( particle, neighbor particle )
    END
  END
  FOR each particle DO
    Update Particle Positions ( particle, particle acceleration )
  END
  Do Anything Useful ( )
END
```

Algorithm. Pseudo-code simulation algorithm

Particle Modeling

Parameters

n	number of particles	scalar
$\overrightarrow{p_i}$, $\overrightarrow{p_j}$	position vectors	vector3
$\overrightarrow{\Delta_{ij}}$	slope vector between particles i and j	vector3
d_{ij}	distance between particles i and j	scalar
$\overrightarrow{a_i}$	acceleration vector of particle i	vector3
m_i , m_j	particles masses	scalar
M	mass factor	scalar
ζ	particles damping factor	scalar

Particle Modeling

Finding a particle acceleration

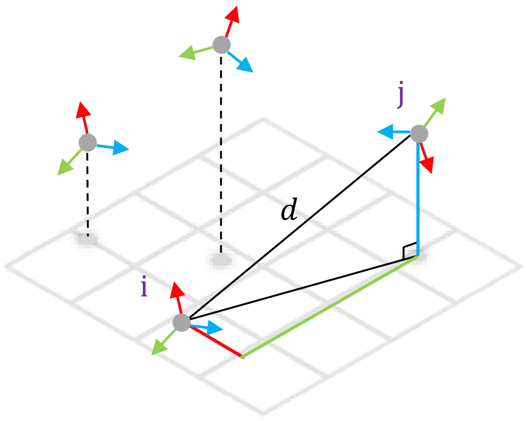


Figure. Two particles i, j and their Euclidean distance d

- $\vec{p_i} = (xi, yi, zi)$
- $\vec{p_j} = (xj, yj, zj)$
- $\overrightarrow{\Delta_{ij}} = \vec{p_j} - \vec{p_i}$ $\quad\quad \overrightarrow{\Delta_{ij}} = \begin{pmatrix} xj - xi \\ yj - yi \\ zj - zi \end{pmatrix}$
- $d_{ij} = \sqrt{(xj - xi)^2 + (yj - yi)^2 + (zj - zi)^2}$
- $\vec{a_i} = \sum_{j \neq i, j=0}^{n-1} \overrightarrow{\Delta_{ij}} \times M \times \zeta \times \frac{1}{{d_{ij}}^3} \times m_j$

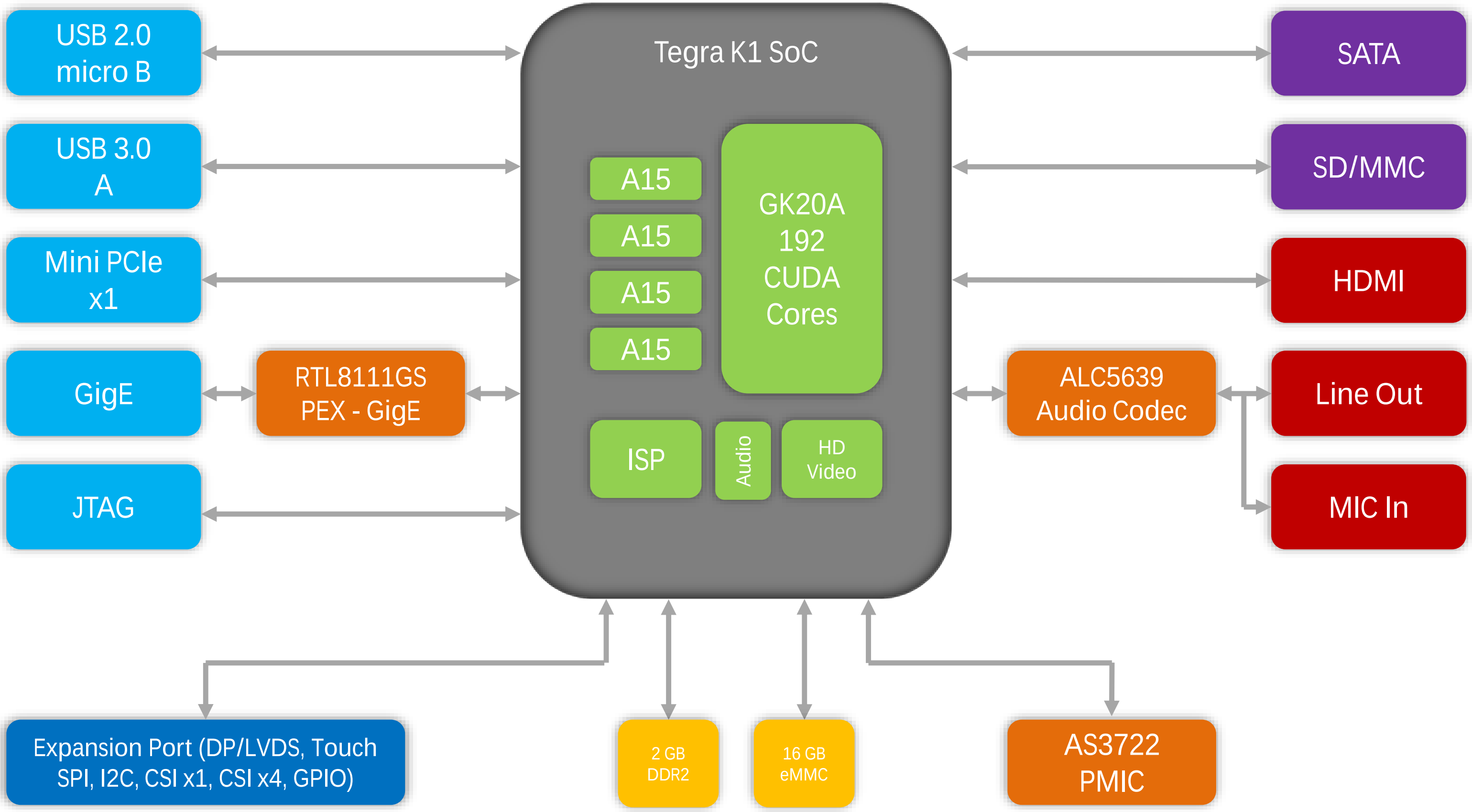
Implementation

targeted hardware

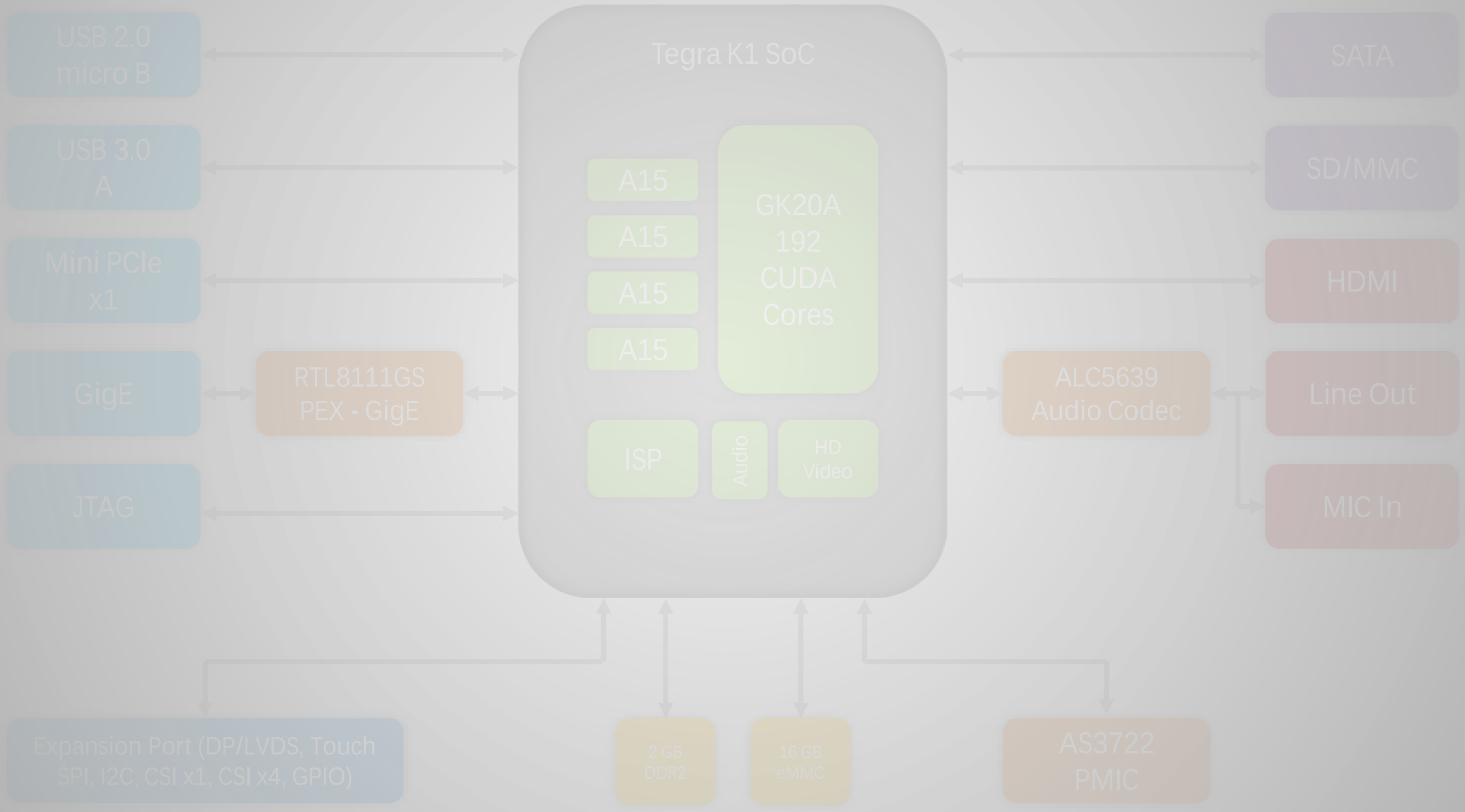
NVIDIA Jetson TK1 development board

http://www.nvidia.com/object/tegra-k1-processor.html
http://www.hardware.fr/focus/imprimer/94/

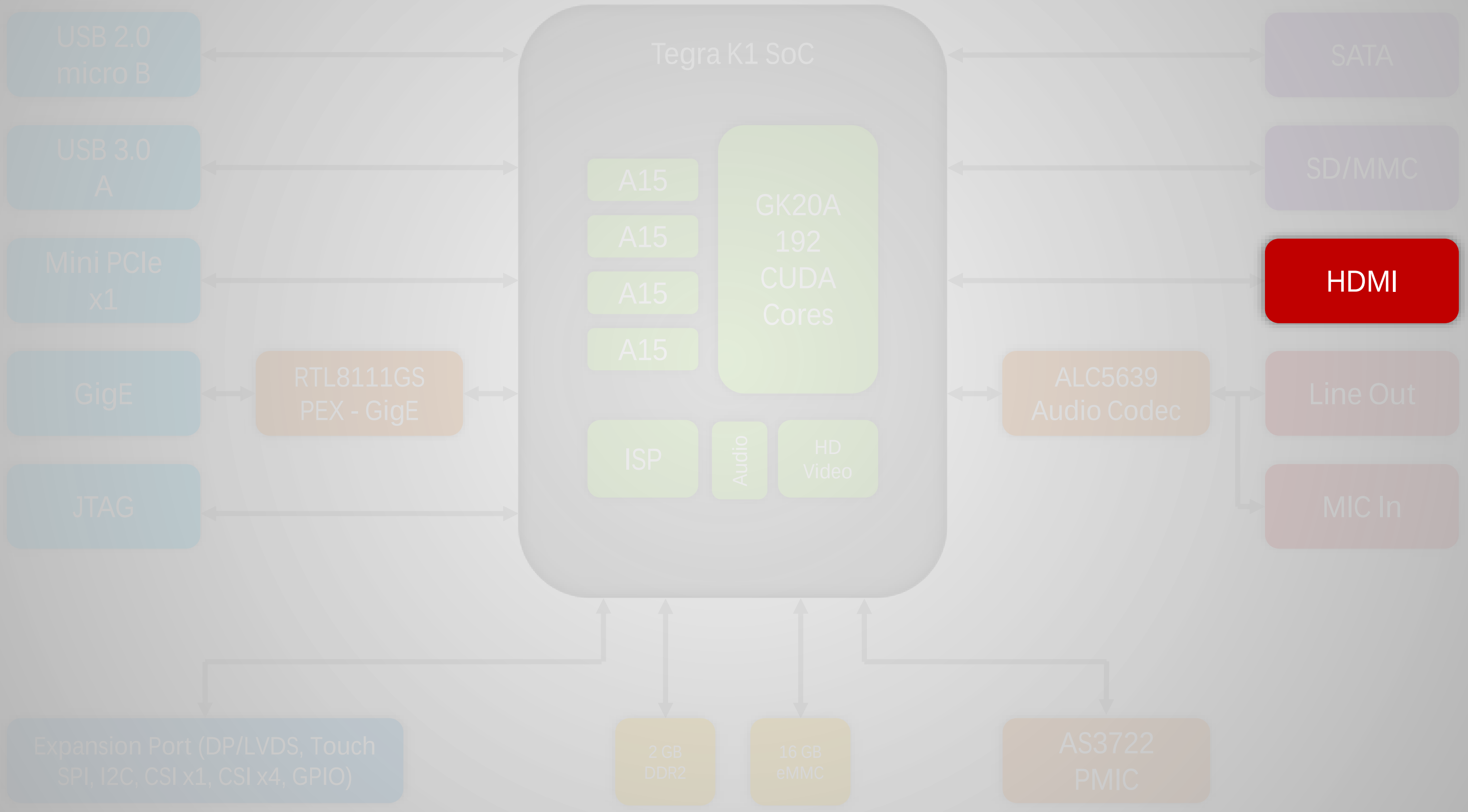
NVIDIA Jetson TK1 development board



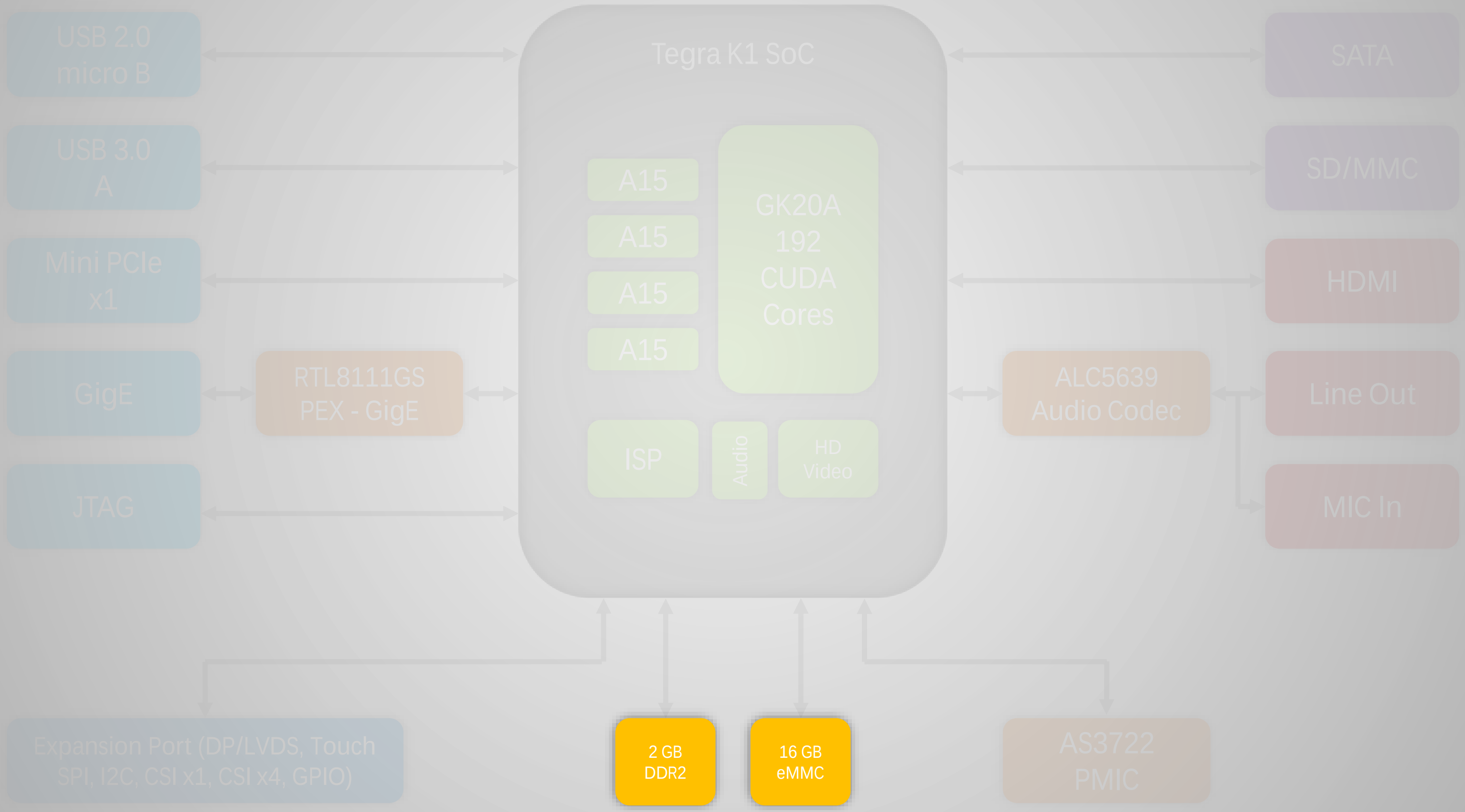
NVIDIA Jetson TK1 development board



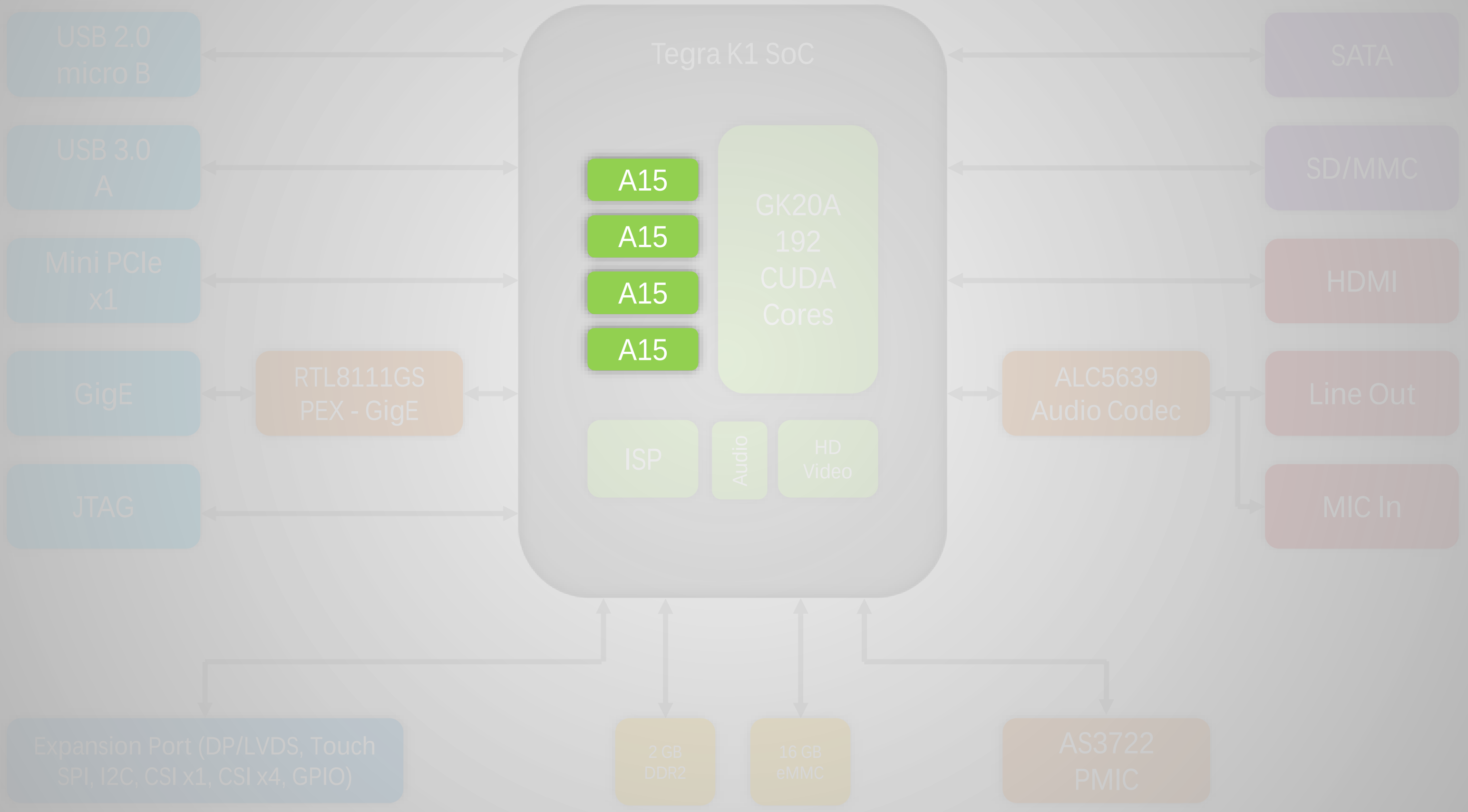
NVIDIA Jetson TK1 development board



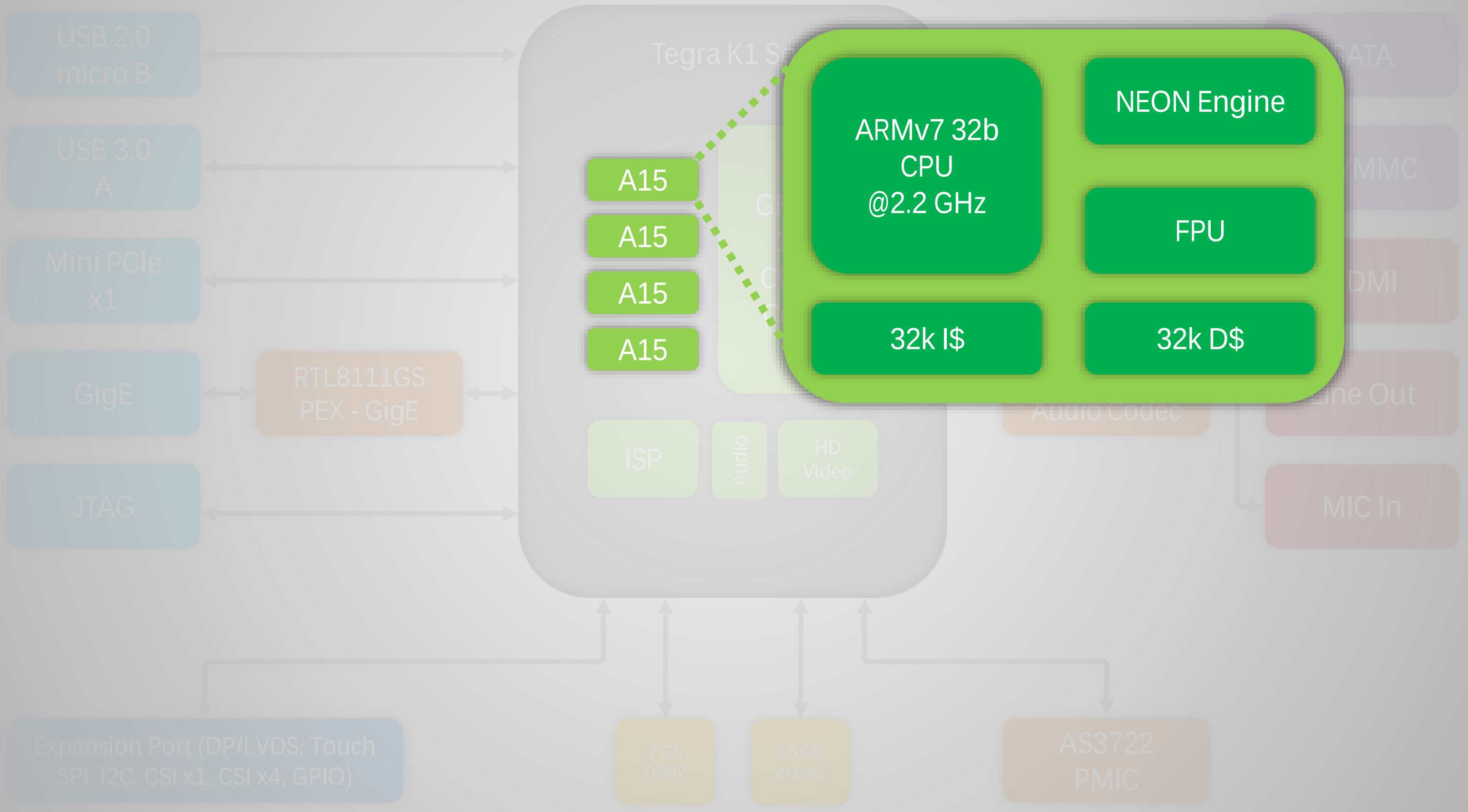
NVIDIA Jetson TK1 development board



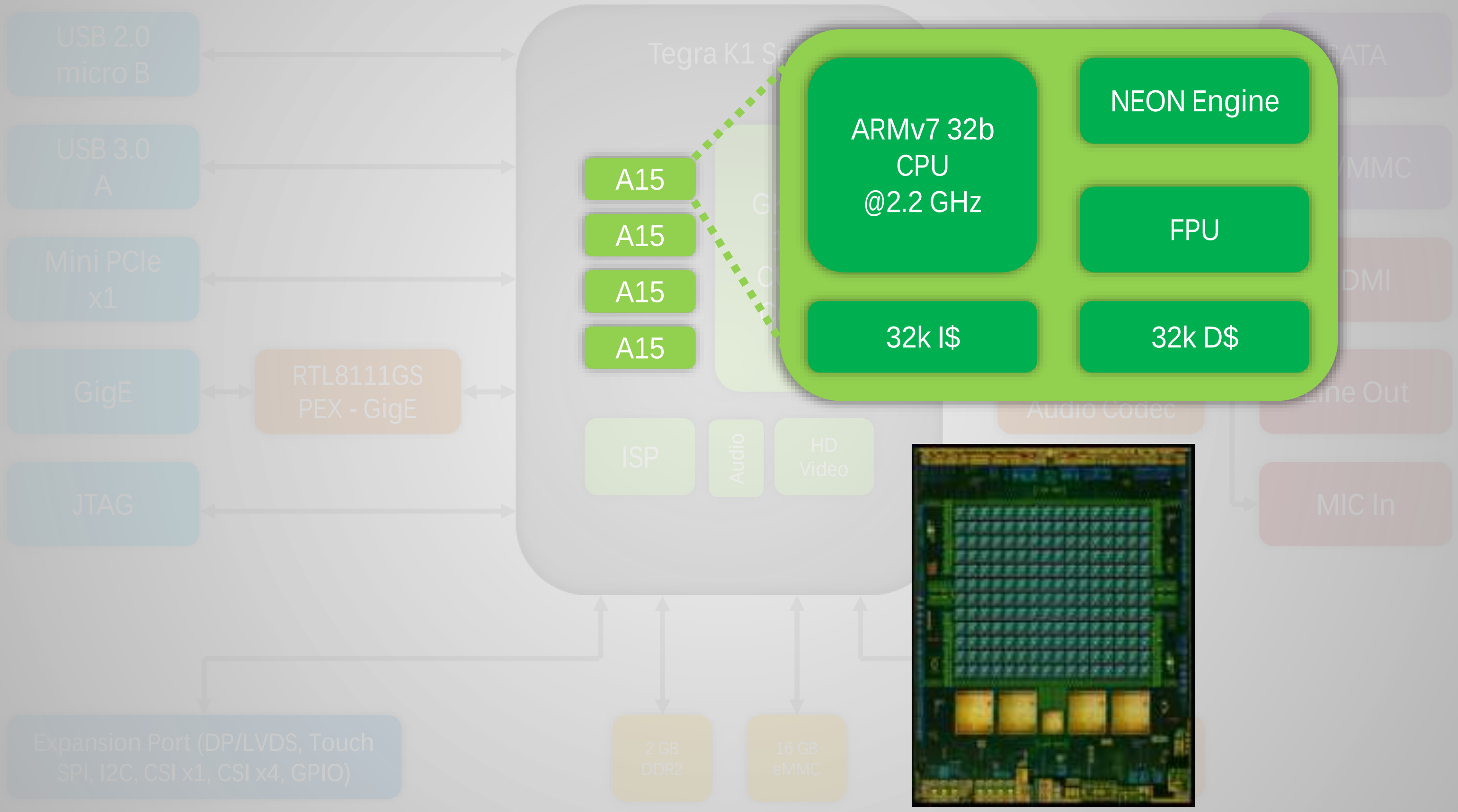
NVIDIA Jetson TK1 development board



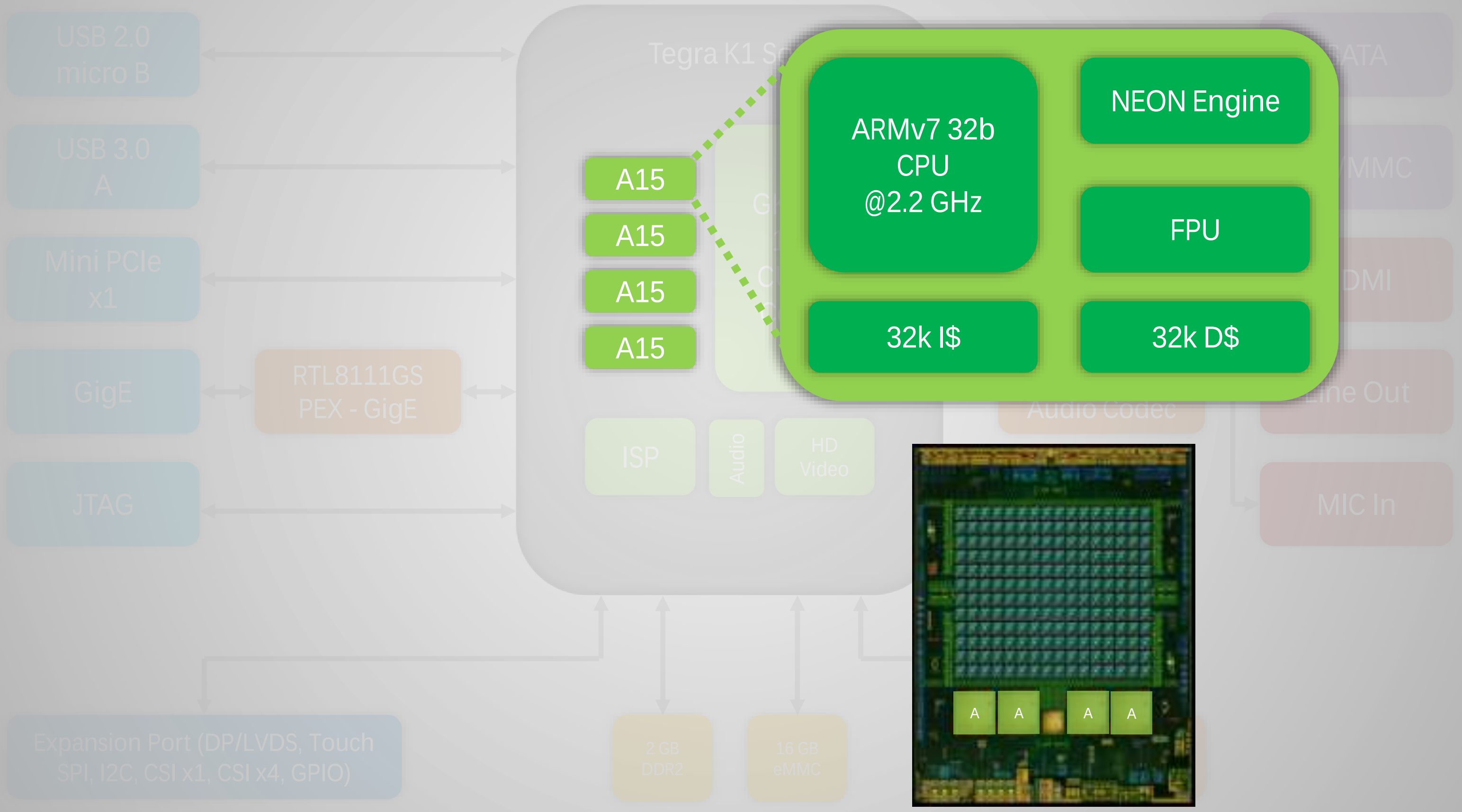
NVIDIA Jetson TK1 development board



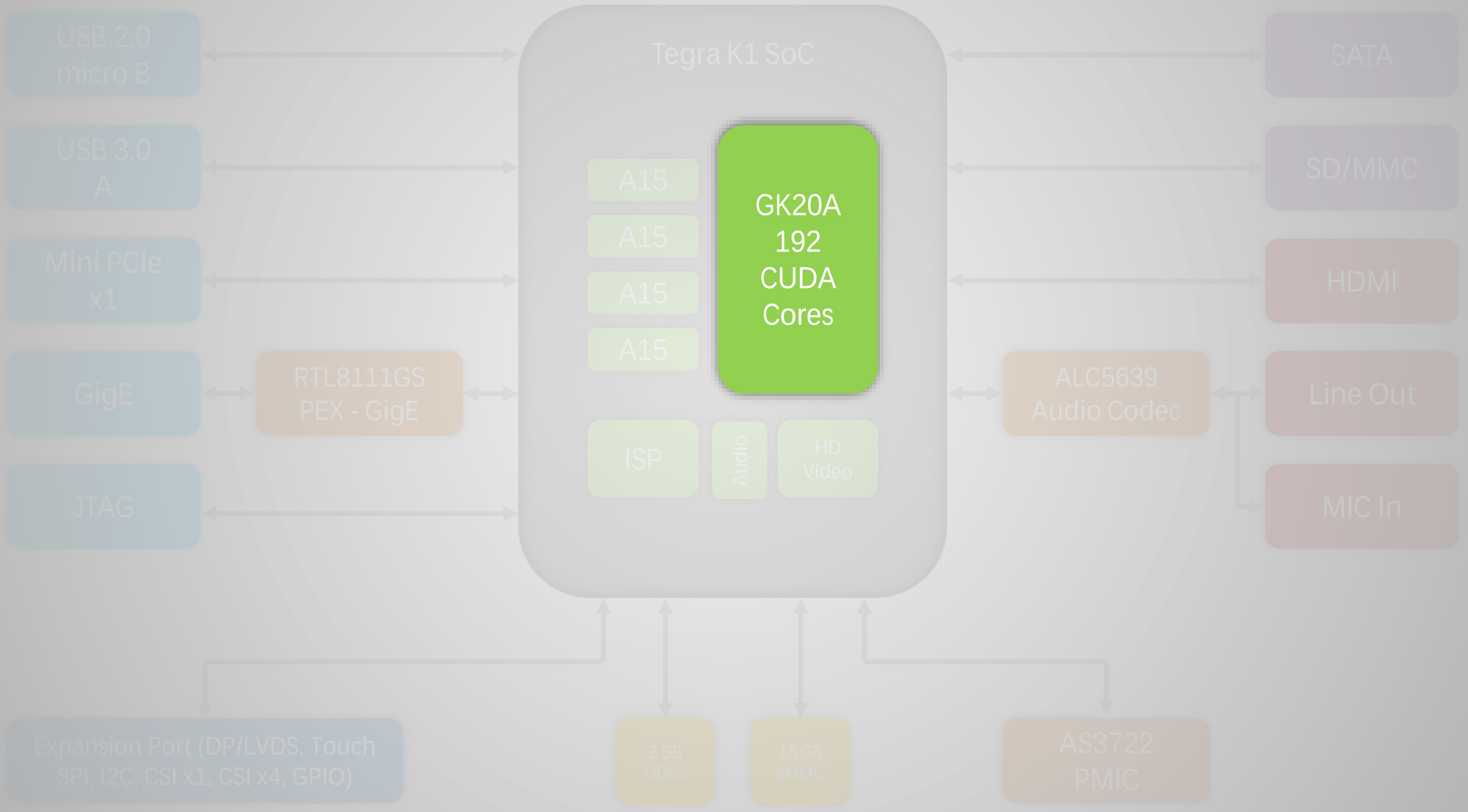
NVIDIA Jetson TK1 development board



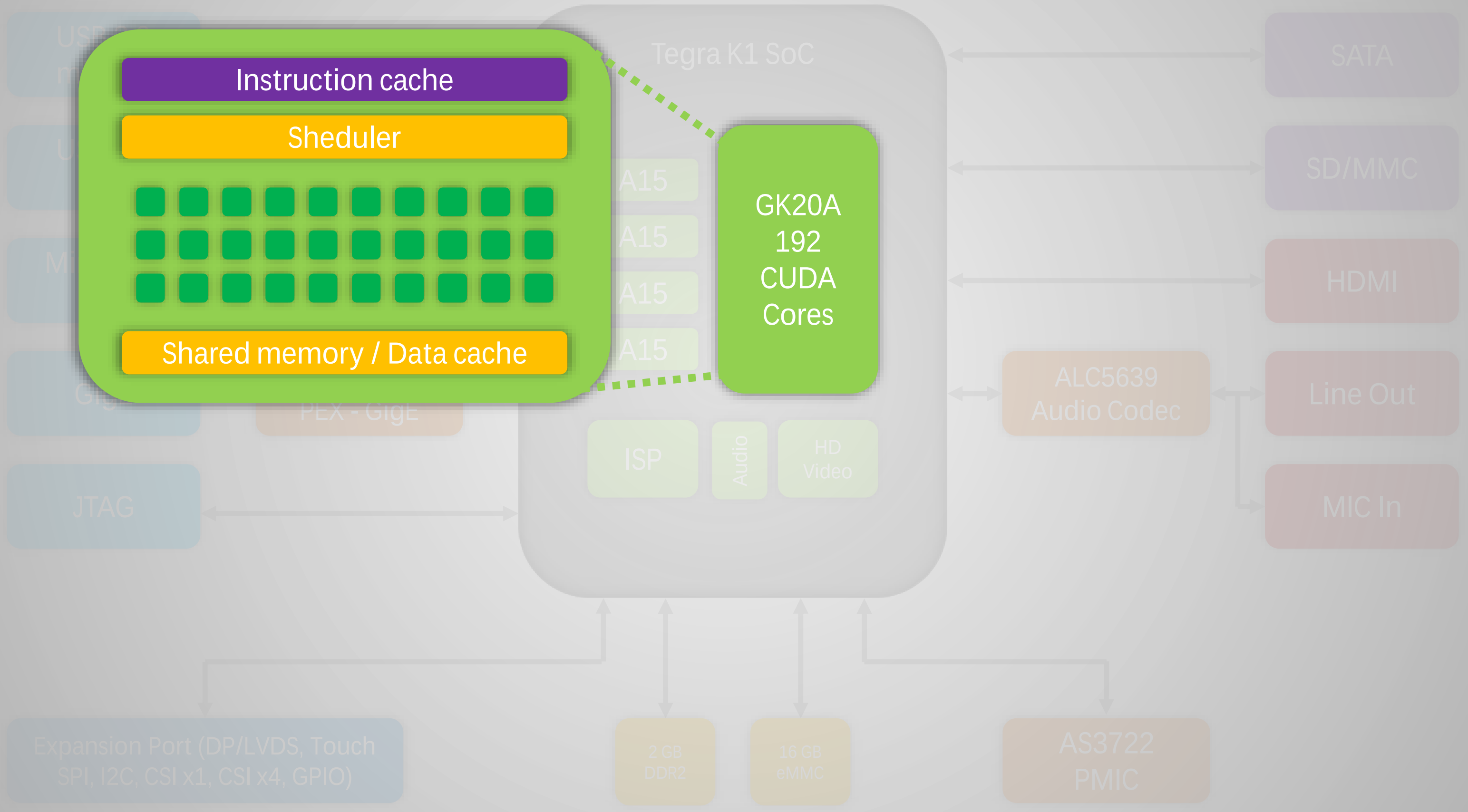
NVIDIA Jetson TK1 development board



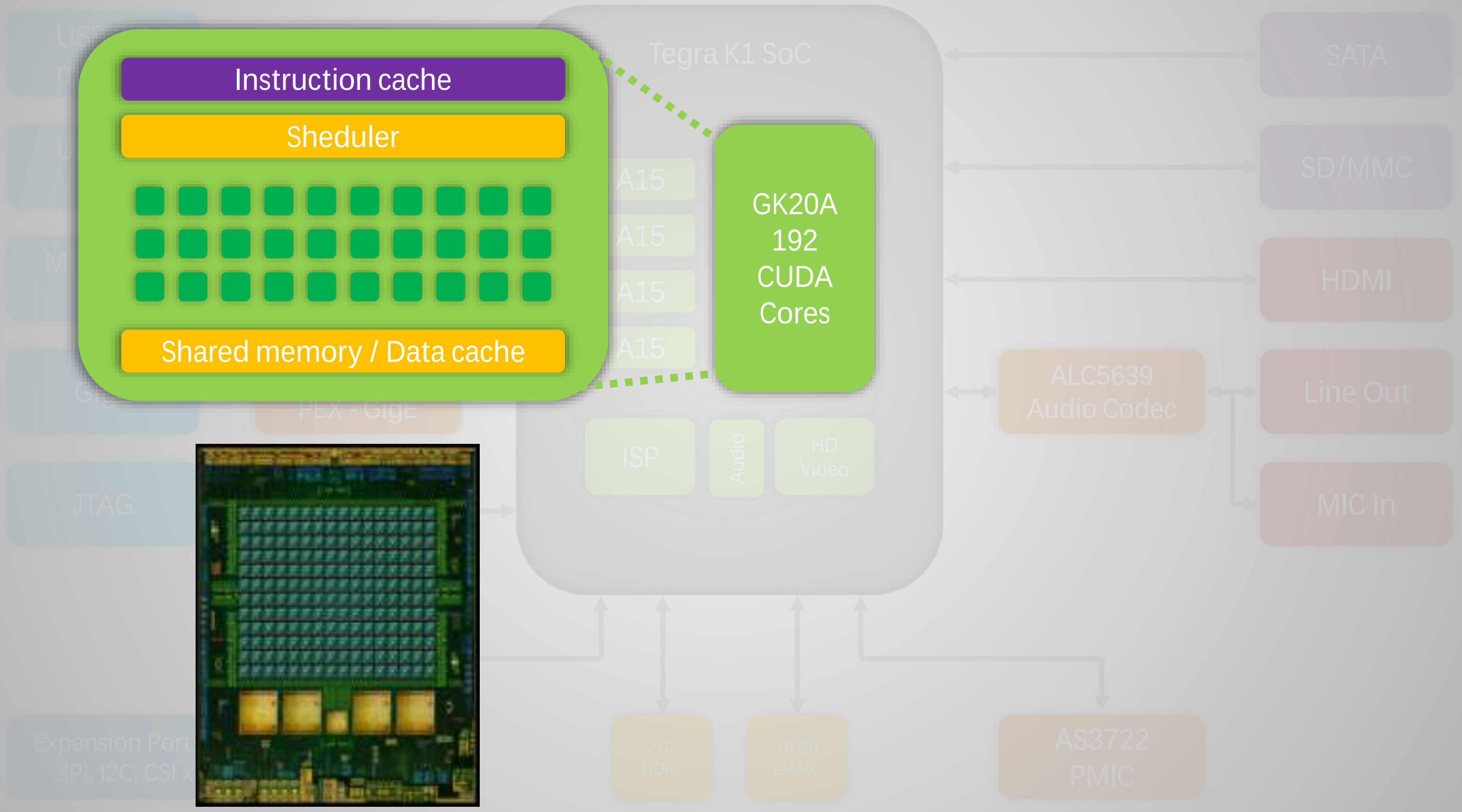
NVIDIA Jetson TK1 development board



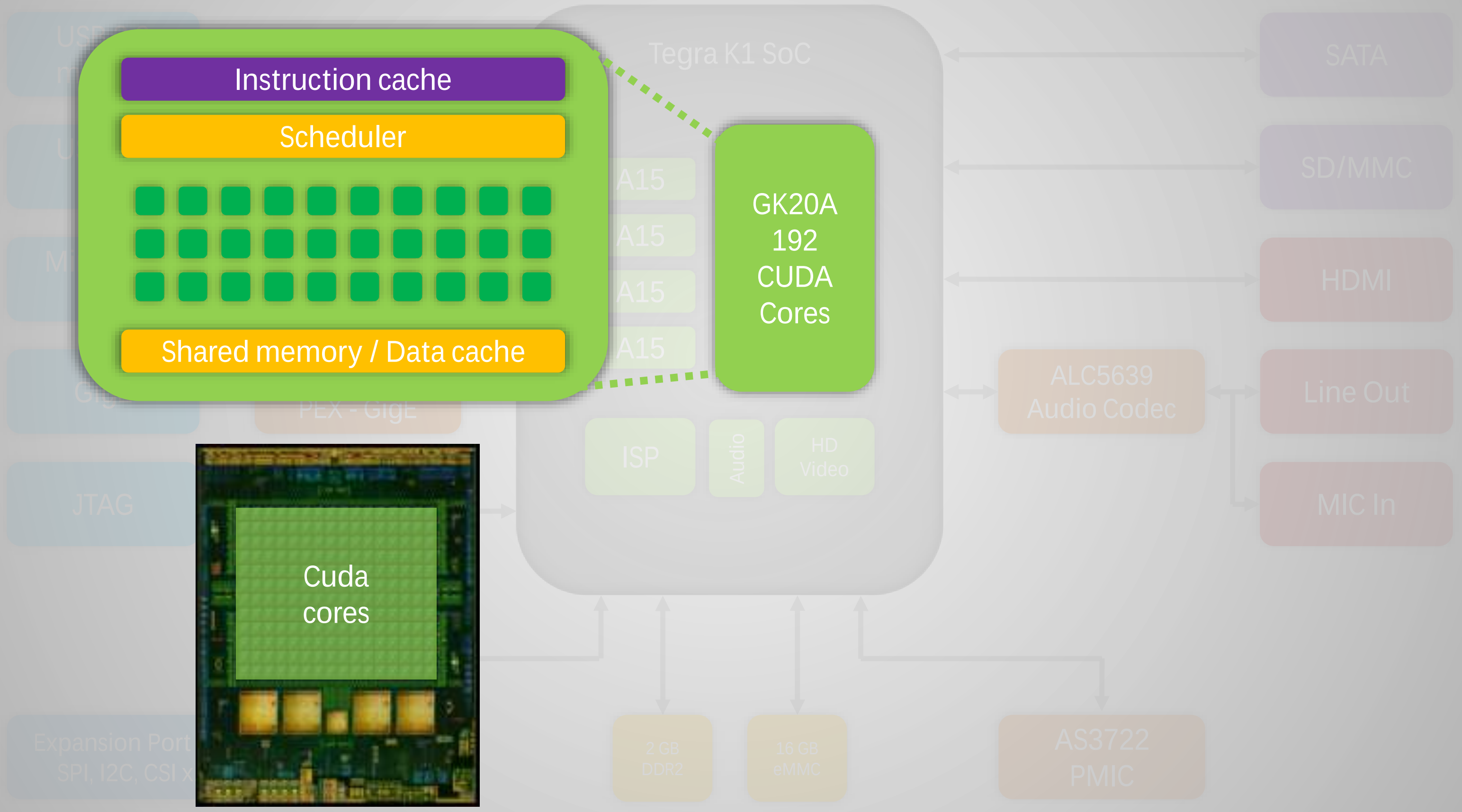
NVIDIA Jetson TK1 development board



NVIDIA Jetson TK1 development board



NVIDIA Jetson TK1 development board



NVIDIA Jetson TK1 development board

Operating system Ubuntu 14.04 LTS

https://developer.nvidia.com/linux-tegra-rel-19

NVIDIA Jetson TK1 development board

Useful compatible libraries

The Open graphics library OpenGL n API for rendering 2D and 3D vector graphics using a GPU.

The OpenGL extension wrangler library GLEW bles OpenGL latest graphics capabilities using extensions.

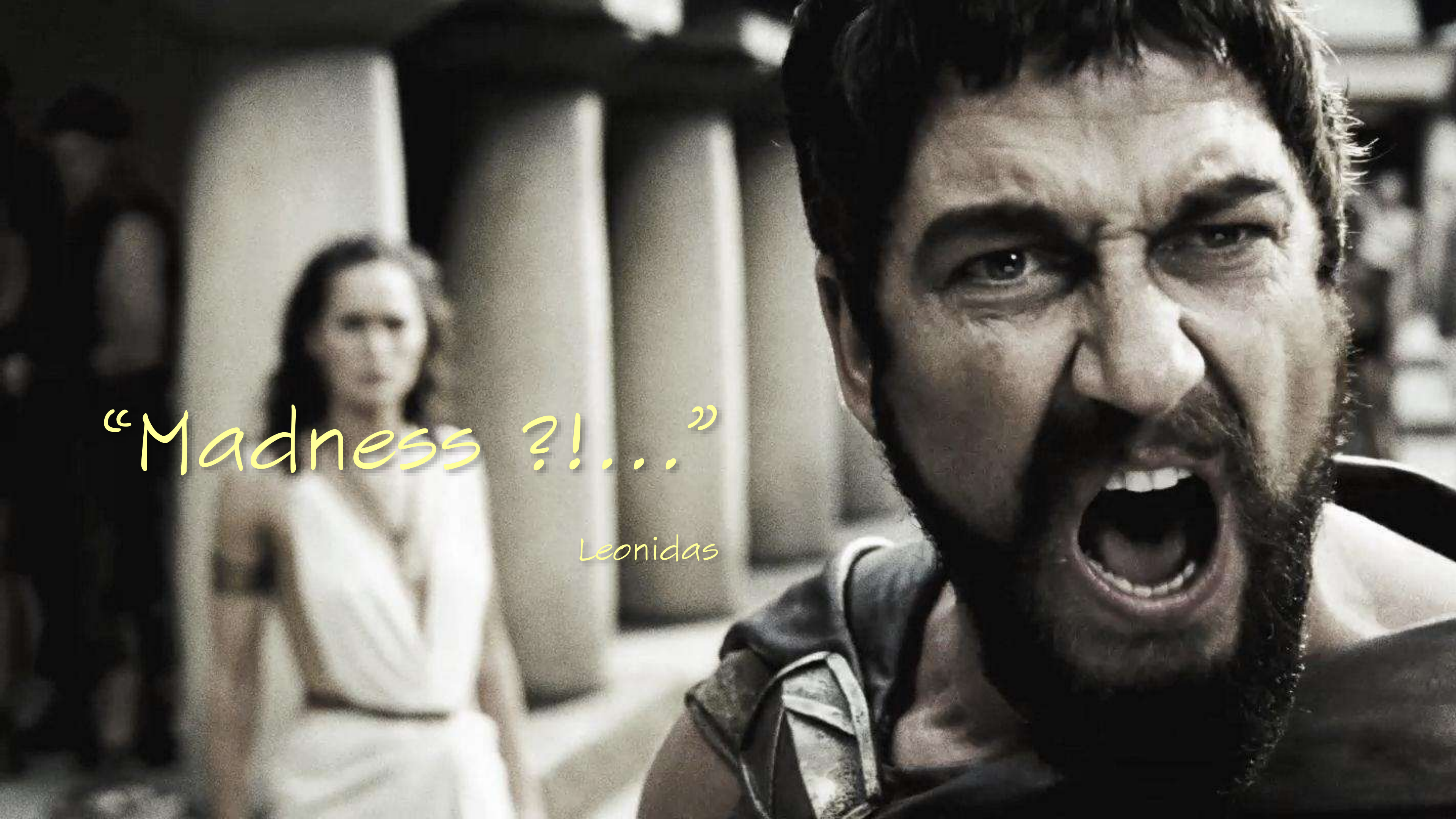
SDL or Simple Directmedia Layer is also a library which provides low access to windows, keyboard and mouse.

The Open Multi-Processing API OpenMP CPUs parallel architectures to unleash relevant application speed-up. To achieve further acceleration, NVIDIA Compute Unified Device Architecture makes uses of NV nVIDIA CUDA GPUs.

NVIDIA Jetson TK1 development board

Useful links

- https://www.opengl.org/
- http://glew.sourceforge.net/
- https://www.libsdl.org/
- http://openmp.org/wp/
- http://www.nvidia.fr/object/cuda-parallel-computing-fr.html
- https://computing.llnl.gov/tutorials/pthreads/


“Madness ?!...”
Leonidas

Where to start ?

Where to start ?

One small step for man...

Boot your board and log in (SE or SEE)
Please take care of it !

Where to start ?

Inside the ./Documents/examples directory

01_helloworld
02_sdl
03_opengl
04_pthread
05_openmp
06_cuda

A set of 6 basic examples to help you getting started

Where to start ?

Inside the ./Documents/examples directory

Where to start ?

Inside the ./Documents/examples directory

01_helloworld

02_sdl

03_opengl

04_pthread

05_openmp

06_cuda

```
> $: cd 01_helloworld
> $: make all
> $: ./bin/helloworld
```

Where to start ?

Inside the ./Documents/examples directory

01_helloworld

02_sdl

03_opengl

04_pthread

05_openmp

06_cuda

```
> $: cd 02_sdl
> $: make all
> $: ./bin/sdl
```

Where to start ?

Inside the ./Documents/examples directory

01_helloworld

02_sdl

03_opengl

04_pthread

05_openmp

06_cuda

```
> $: cd 03_opengl
> $: make all
> $: ./bin/opengl
```

Where to start ?

Inside the ./Documents/examples directory

01_helloworld 02_sdl 03_opengl

04_pthread 05_openmp 06_cuda

```
> $: cd 04_pthread
> $: make all
> $: ./bin/pthread
```

Where to start ?

Inside the ./Documents/examples directory

01_helloworld   02_sdl   03_opengl

04_pthread   05_openmp   06_cuda

```
> $: cd 05_openmp
> $: make all
> $: ./bin/openmp
```

Where to start ?

Inside the ./Documents/examples directory

01_helloworld

02_sdl

03_opengl

04_pthread

05_openmp

06_cuda

```
> $: cd 06_cuda
> $: make all
> $: ./bin/cuda
```

Where to start ?

Milky way and andromeda galaxies data set

Go to this page :
http://bima.astro.umd.edu/nemo/archive/#dubinski

and download John Dubinski's archive :
dubinski.tab.gz

John Dubinski

Where to start ?

John Dubinski's file format

Num. of particles	Total particles	Galaxy	Element
16384	16384	Milky way	disk
16384	32768	Andromeda	disk
8192	40960	Milky way	bulge
8192	49152	Andromeda	bulge
16384	65536	Milky way	halo
16384	81920	Andromeda	halo

The text file describes each particle line by line. Each line is a string that follows this format, where space is used as delimiter :

Mass PositionX PositionY PositionZ VelocityX VelocityY VelocityZ

Video time

"Just one more thing..."

Columbo

Just one more thing...

Q & A

Q: We need to open a new window and we want to use my mouse and my keyboard to play with. How do I do that ?

A: Use SDL, it's not the only available library to do it, but at least it's a good one

Just one more thing...

Q & A

Q: How do we visualize our particles in a 3D space ?

A: OpenGL is the key. There is a basic example for manipulating 3D in your ./Documents/examples directory

Just one more thing...

Q & A

Q: OK, We're able to draw points in 3D, now what ?

A : See if everything is correct. Download John's Dubinsky data set and see what's happening. If it looks good, then you can start playing with maths :)

Just one more thing...

Q & A

Q: Our simulation is working but is slow as a Romero's zombie, is it normal ?

A : Yes, it is. But you can use OpenMP or/and CUDA to speed-up things

Just one more thing...

Q & A

Q: What if we succeed in this project ?

A : Well, then you could say ...


"We came, we saw, we kicked its ass!"
Dr. Peter Venkman