



Information Coding / Computer Graphics, ISY, LiTH

The Fast Fourier Transform on GPUs

Some images in this part by Mario Garrido



Information Coding / Computer Graphics, ISY, LiTH

The Fast Fourier Transform (FFT)

Fast implementation of the Fourier Transform

Converts a signal to frequency space

Very important algorithm in signal processing



FFT

**Computes the Discrete Fourier Transform (DFT)
of a signal of N samples in $N \log N$ time**

**Many variants. Cooley-Tukey (1965) most
common.**



DFT

The Discrete Fourier Transform

Converts a signal to frequency space

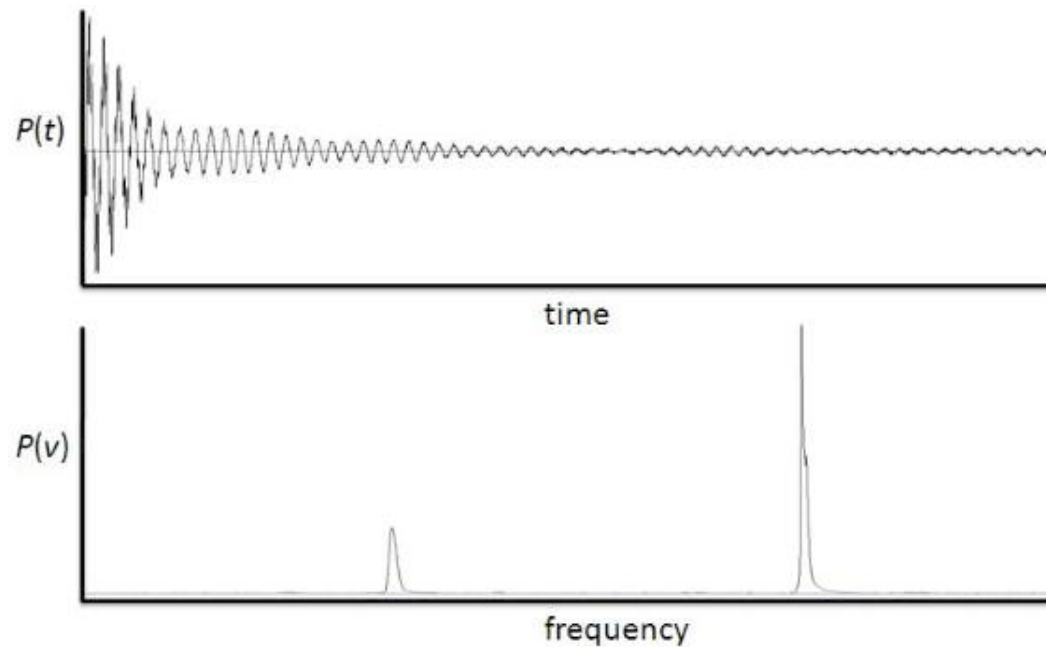
Essentially a series of convolutions with harmonic functions of varying frequency

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi k \frac{n}{N}} \quad k = 0, \dots, N - 1.$$



DFT example

1D signal to frequency space (e.g. sound)





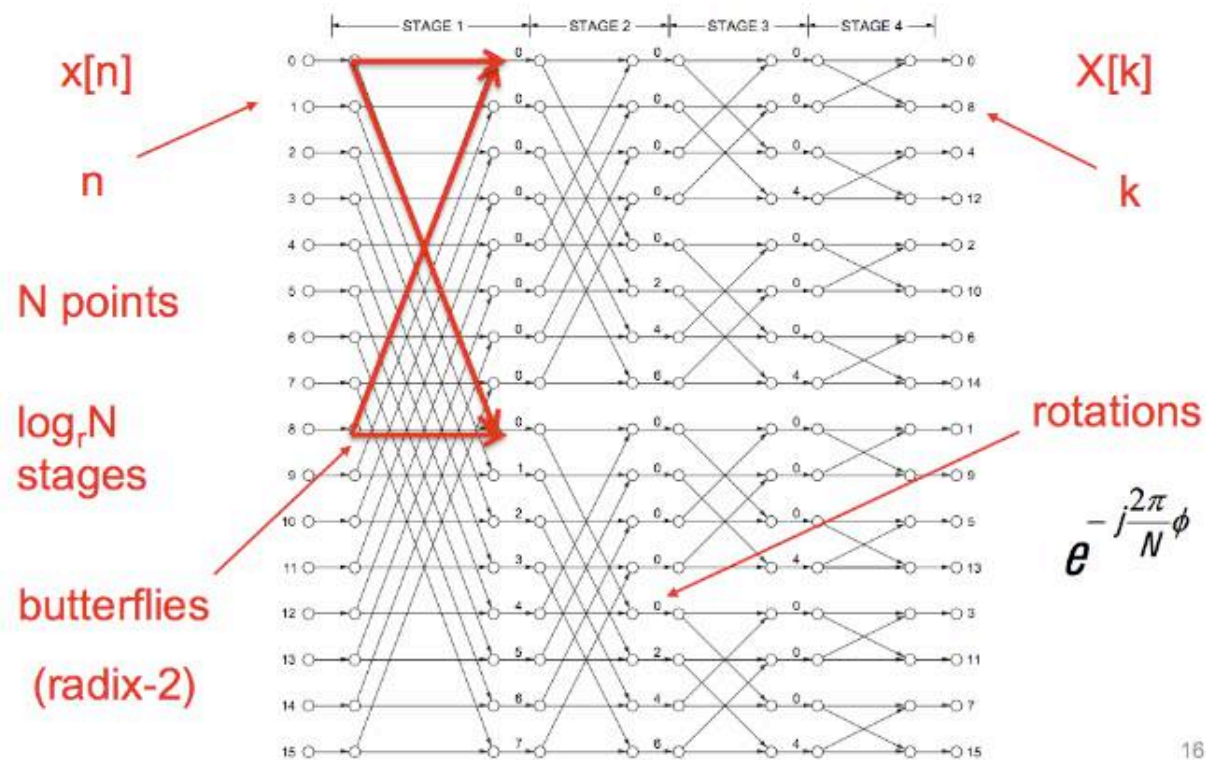
DFT example

2D signal to frequency space (e.g. images)





FFT flow graph (Radix-2)





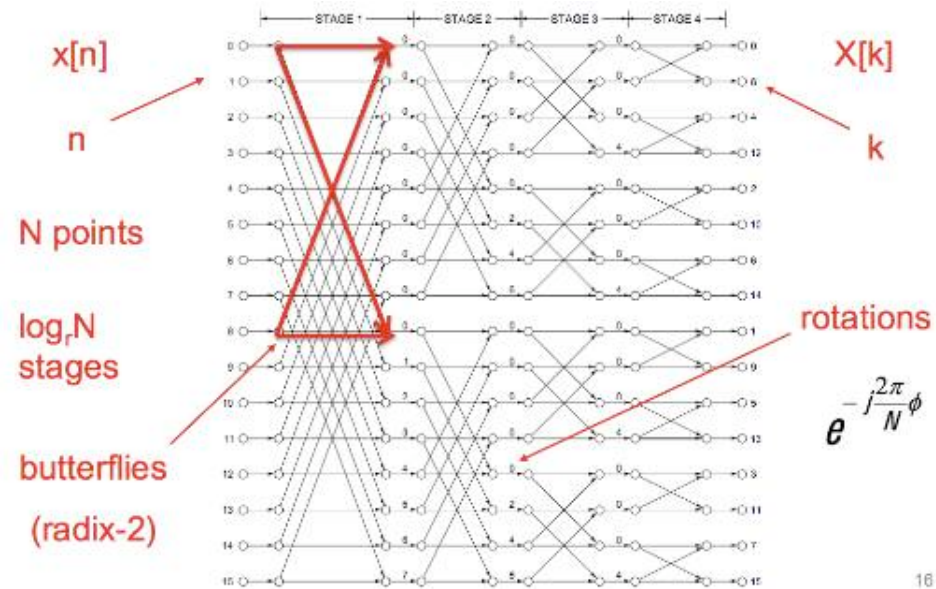
FFT in parallel

Pretty parallel from the start!

**BUT very large jumps in memory for
some stages!**



Information Coding / Computer Graphics, ISY, LiTH



"Large" stages:
Can not be
performed within
shared memory!

"Small" stages:
Can be performed
within shared
memory!



Possible approach

Perform all "small" stages in a single run, using shared memory. Very fast!

Perform all "large" stages as separate kernel runs.



Information Coding / Computer Graphics, ISY, LiTH

NVidia "made your bed" for FFT

cufft, CUDA FFT, included in all CUDA distributions

A well optimized CUDA implementation



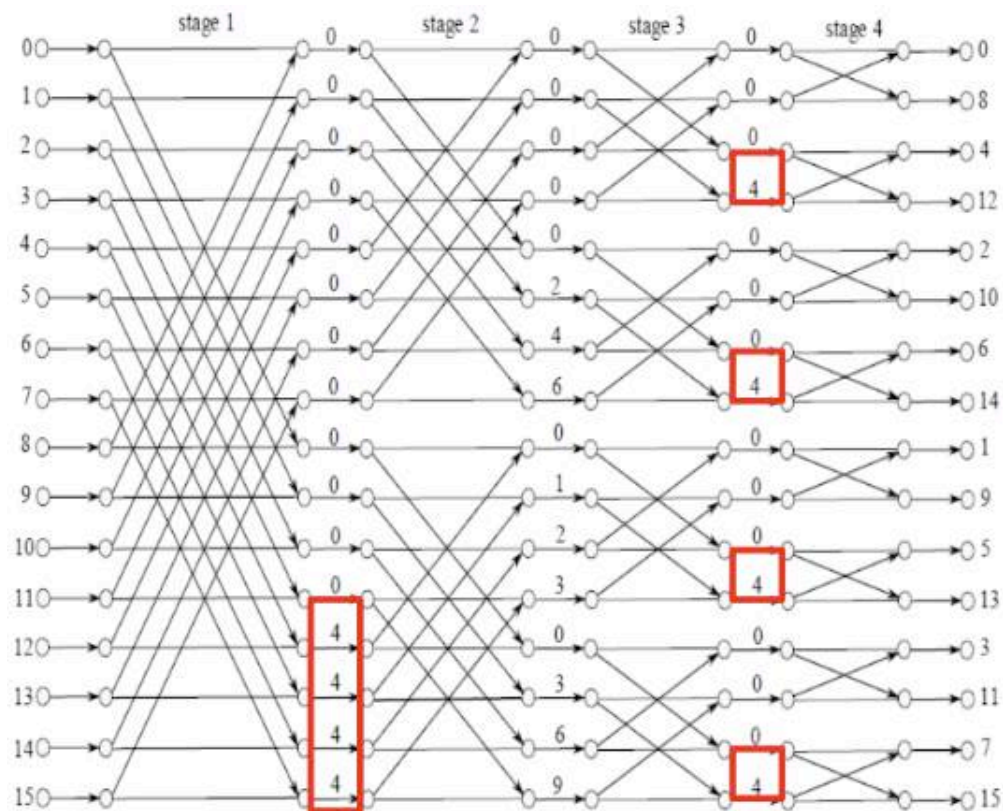
But there are alternatives!

**Optimization approaches made in a
specific implementation**

2013 publication: "New Radix-2 and Radix-2² Constant
Geometry Fast Fourier Transform Algorithms for GPUs",
Ambuluri, Garrido, Ogniewski, Ragnemalm, Caffarena

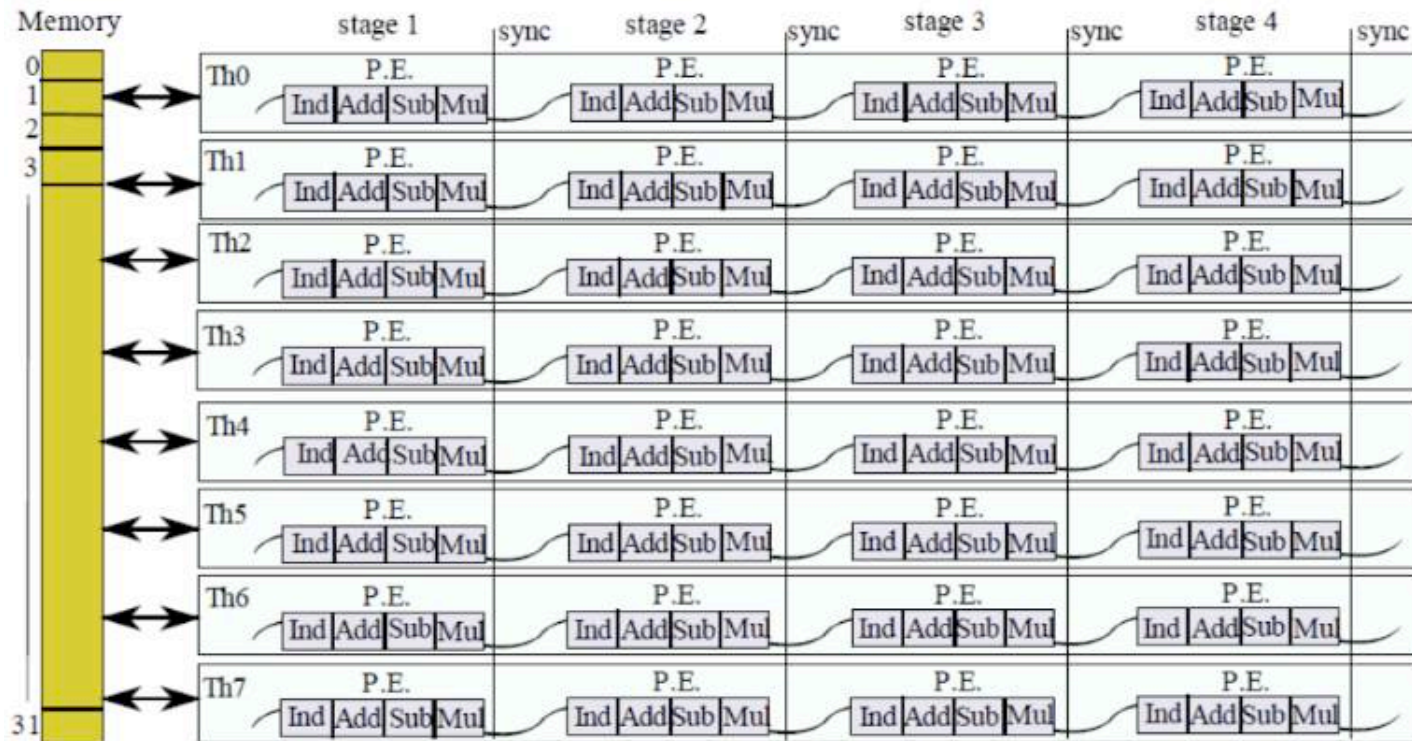


Simplify - use Radix-2²



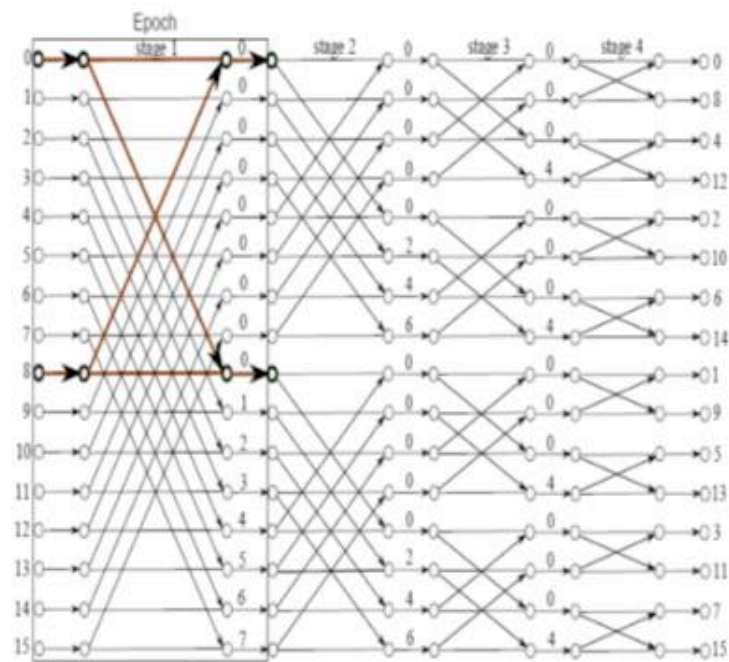


Use shared memory

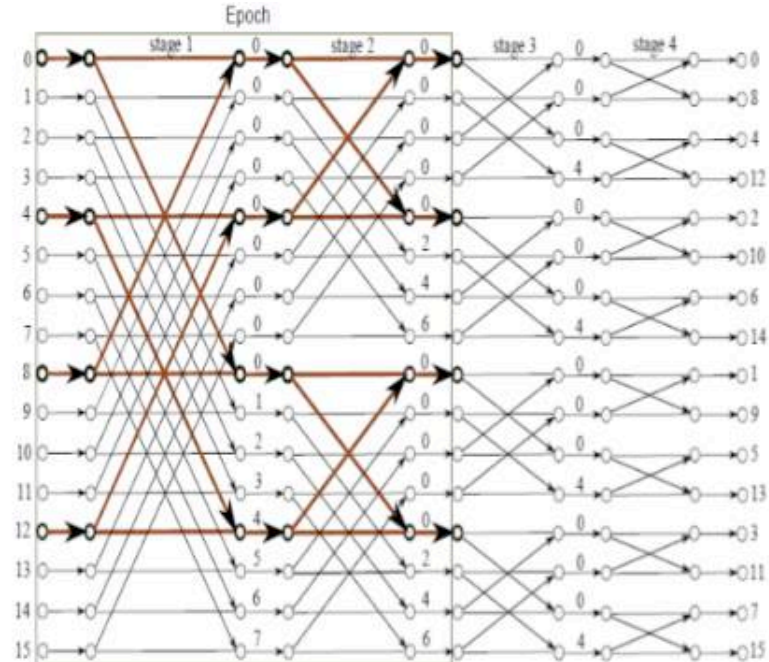




Reduce synch. points



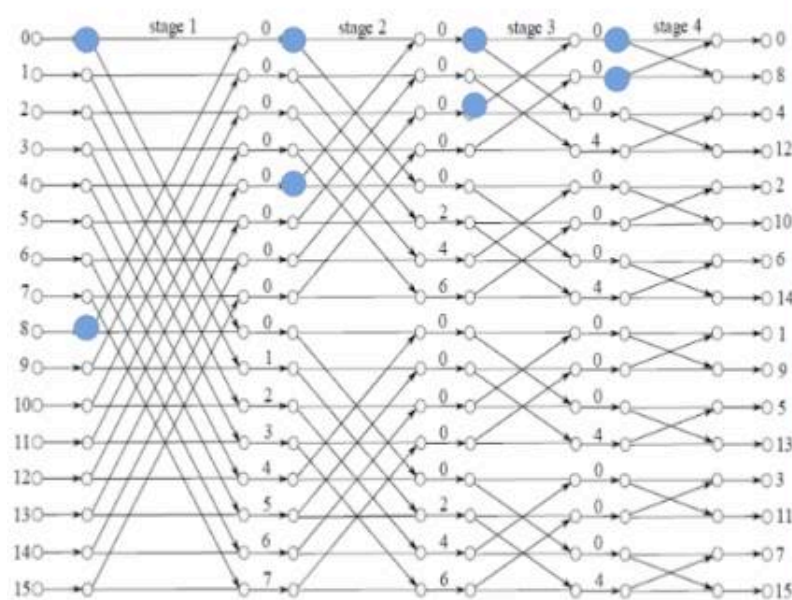
2-word group



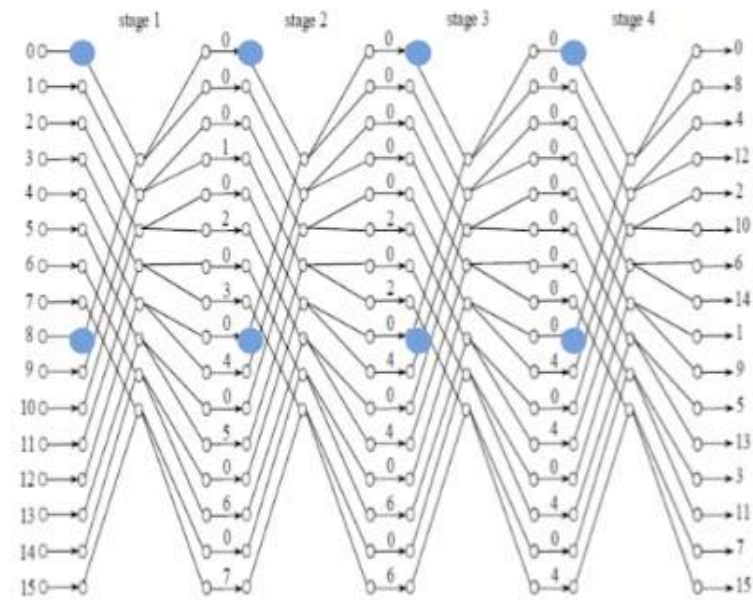
4-word group



Reduce index calculations: Constant geometry FFT



Conventional flow graph

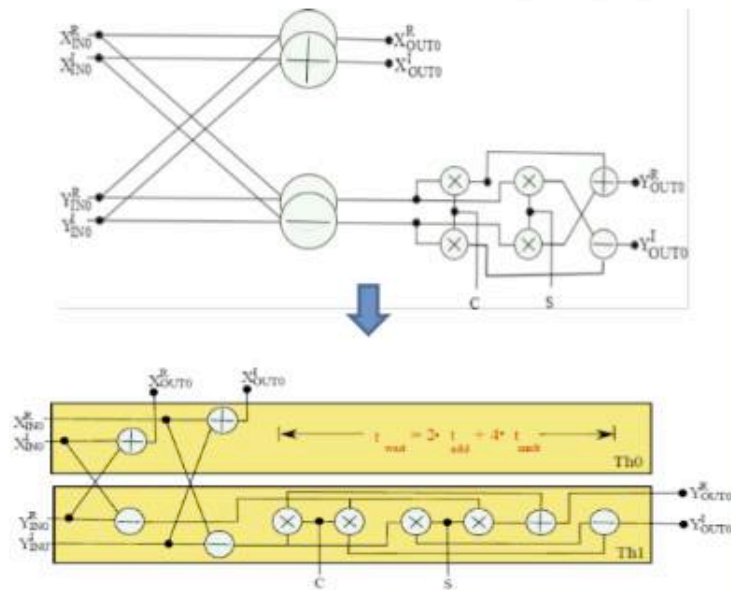


Constant Geometry

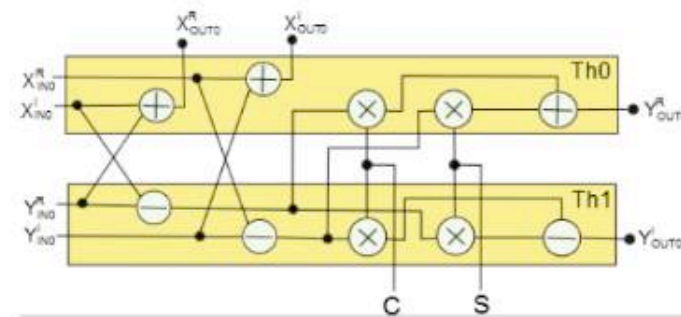


Balance load between threads

USE SCHEDULING



Unbalanced scheduling



Balanced scheduling



Information Coding / Computer Graphics, ISY, LiTH

Result:

**Our implementation was significantly faster than
NVIDIA's cufft - that is, for the sizes we tried**

Best paper award at the conference

**Algorithms can often be modified more than it
seems**