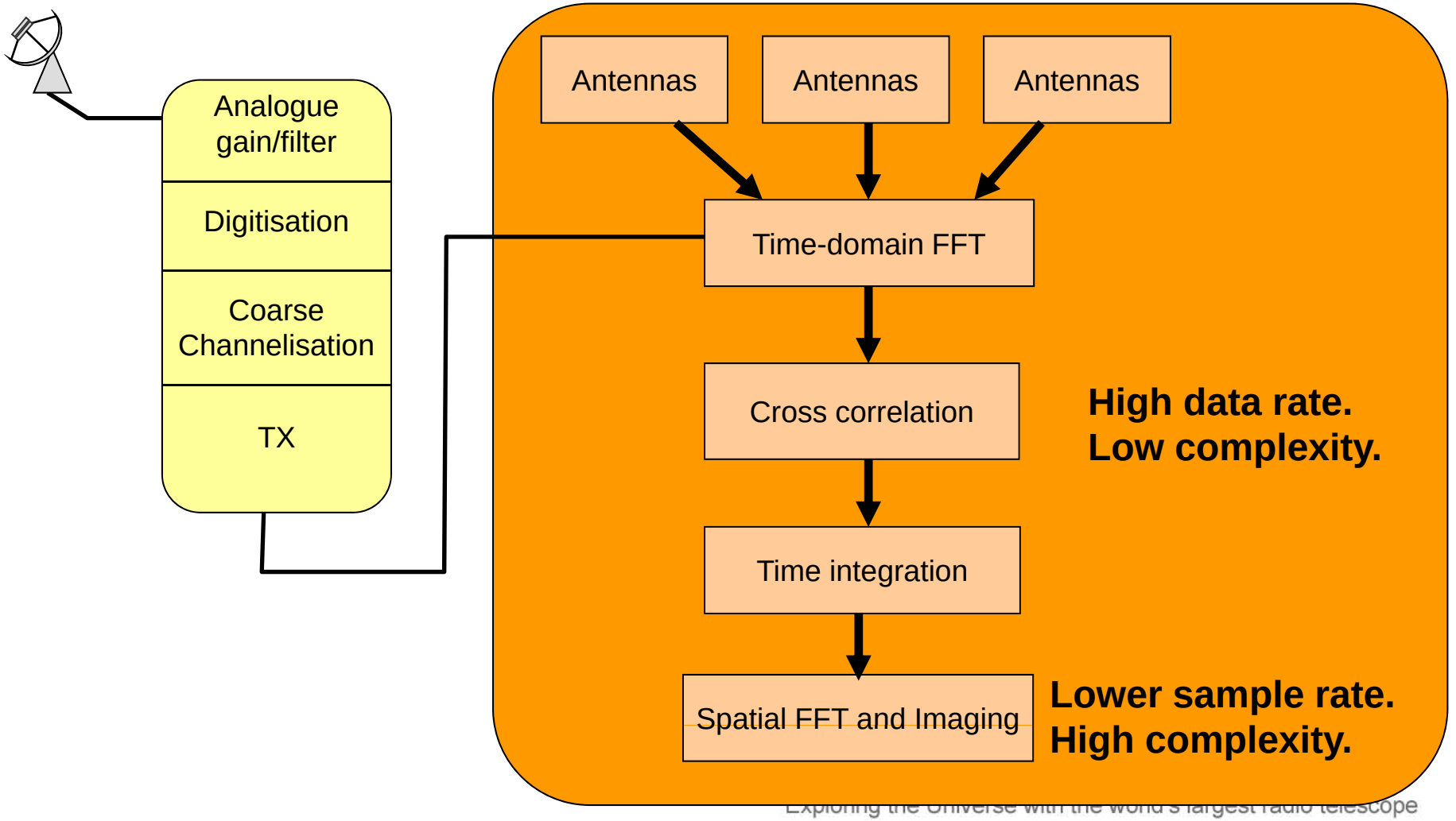

# Software Correlator Concept Description

**Dominic Ford**, University of Cambridge

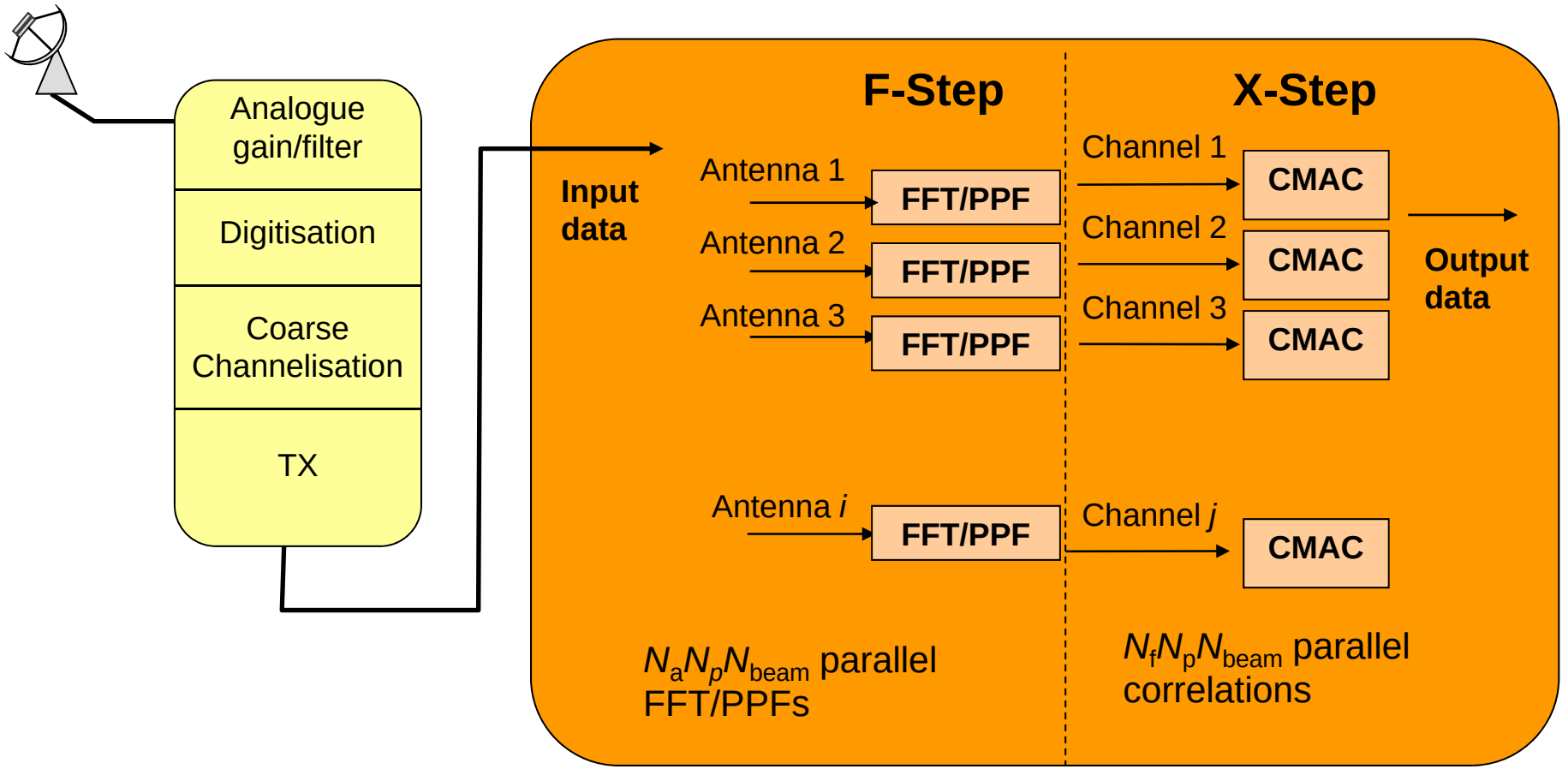
**Andrew Faulkner, Jongsoo Kim, Paul Alexander**

**14th April 2011**

# Interferometer Block Diagram



# Interferometer Block Diagram



# Correlator architectures

- **Hardware correlators**

  FPGA-based WIDAR correlators used by the eVLA and eMERLIN.
  Also, used until recently by the GMRT.
  Will be used by the full 512-antenna MWA.
  **High energy efficiency. Longer development time.**

- **Software correlators**

  Beowulf clusters used by the Australian LBA and VLBA.
  IBM BlueGene used by LOFAR.
  Custom-designed cluster used by GMRT.
  GPGPU system trialled by the 32-antenna prototype MWA.
  **Reduced energy efficiency. Huge gain in flexibility.**

# Advantages of Software Correlators


- **Rapid development cycles**

  Software correlators use conventional programming languages for which mature development tools and debugging suites already exist.

  Can track advances in technology very closely.

- **Reduced NRE**

  Pre-existing hardware is used, which is already mass-produced.

- **Easy reconfigurability**

  A software correlator can be reconfigured post-deployment to use new algorithms, or in response to hardware failure.

  Can use same hardware for other tasks, e.g. beamforming.

# Potential processing architectures

- **Intel massively-parallel x86 chips**

  32-core processor demonstrated in 2010 *(Knight's Bridge);* 50-core processor expected to be demonstrated in June 2011 *(Knight's Corner).*

- **Massively-parallel embedded processors**

  e.g. Picochip or Tilera.

- **Graphics processing units (GPUs)**

  Produced by ATI (part of AMD); latest generation of *Fusion* processors.

  Also produced by NVIDIA; *Tesla* cards specifically designed for HPC.

# NVIDIA Tesla Cards

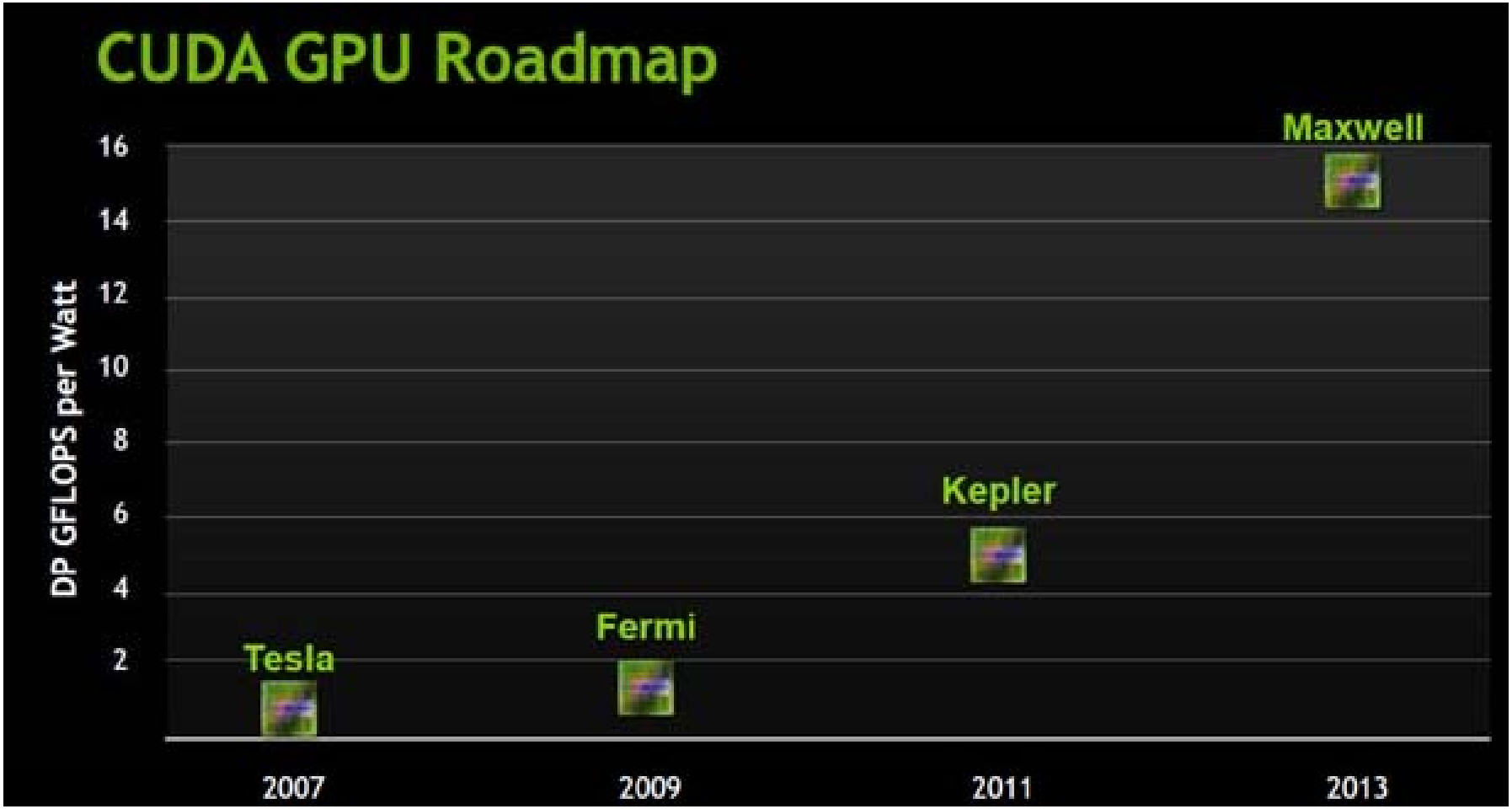
**An NVIDIA *Tesla* card (*Fermi* series, 2009) can deliver 1030.4 GFLOP peak performance.**


**Power dissipation 250W.**

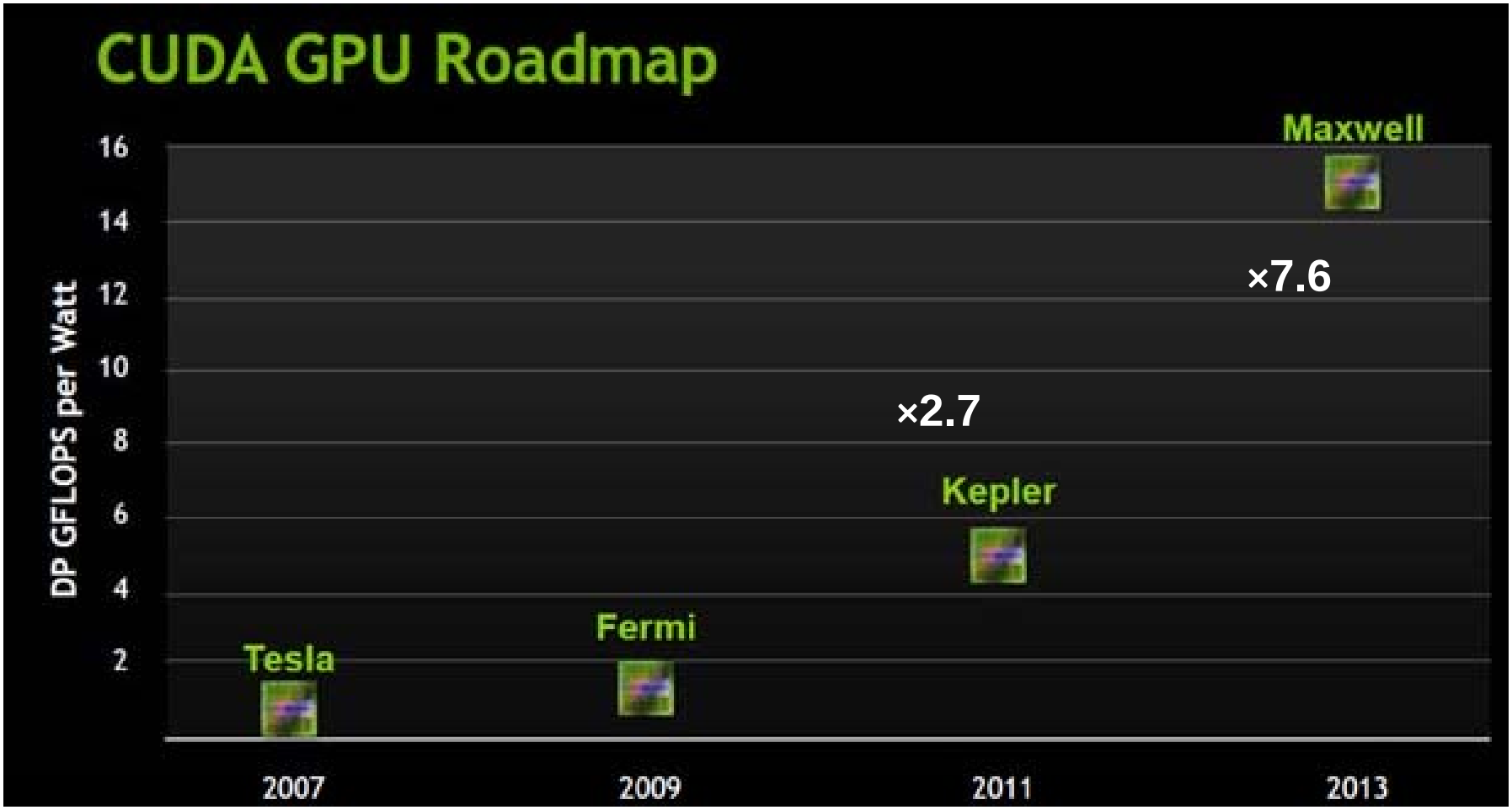
**Data transfer via PCI-Express bus at 64 Gbit/s.**

**But architecture requires highly-parallel code, and typically real applications only achieve around 30% processor utilisation.**

# NVIDIA Tesla Card Roadmap



# NVIDIA Tesla Card Roadmap



# NVIDIA Tesla Card Roadmap

Projecting these numbers forward to the timeframe of $SKA_1$:

| GPGPU card | Expected release year | Performance |
|---|---|---|
| Fermi | 2009 | 1.0× |
| Kepler | 2011 | 2.7× |
| Maxwell | 2013 | 7.6× |
| ??? | 2015 | 15.2× |
| ??? | 2017 | 30.4× |

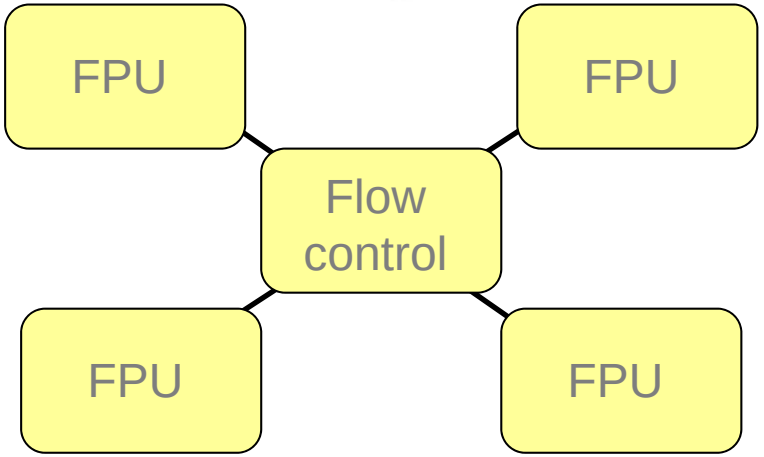
i.e. GPU cards may be expected to deliver 15-30 TFLOP by 2016-2018.

Power dissipation expected to remain constant at 250W per card.

# Making Efficient Use of GPUs

GPUs achieve their outstanding performance by reducing the amount of silicon dedicated to flow control.

GPUs need careful programming to ensure that parallel threads follow common execution paths.




In many applications, this means that very low efficiencies (20-30%) are achieves. For radio astronomy imaging processing, efficiencies as low as a few percent have been reported.

But cross-correlation is **simple** and high efficiencies can be achieved. Greenhill *et al*. report achieving **79% efficiency** with Tesla Fermi cards. I assume **75%** efficiency here.

# FLOP per bit ratio

Tesla Fermi cards can deliver 1030.4 GFLOP.
But, maximum rate of transfer onto card is 64 Gbit/s (PCI-Express 2).

To make efficient use of GPU, need to perform

**> 16 floating point operations per bit**.

c.f. 0.1 FLOP/bit for a typical CPU in 2011.

GPUs are FLOP-heavy processors

# FLOP per bit ratio

## F-step

$g_{\text{F-FLOP}}/g_{\text{F-in}} = (3/2)\log_2 N_f / (2N_b)$

| FFT size | Ratio |
|---|---|
| 1,024 | 1.875 |
| 32,768 | 2.812 |
| 380,000 | 3.475 |

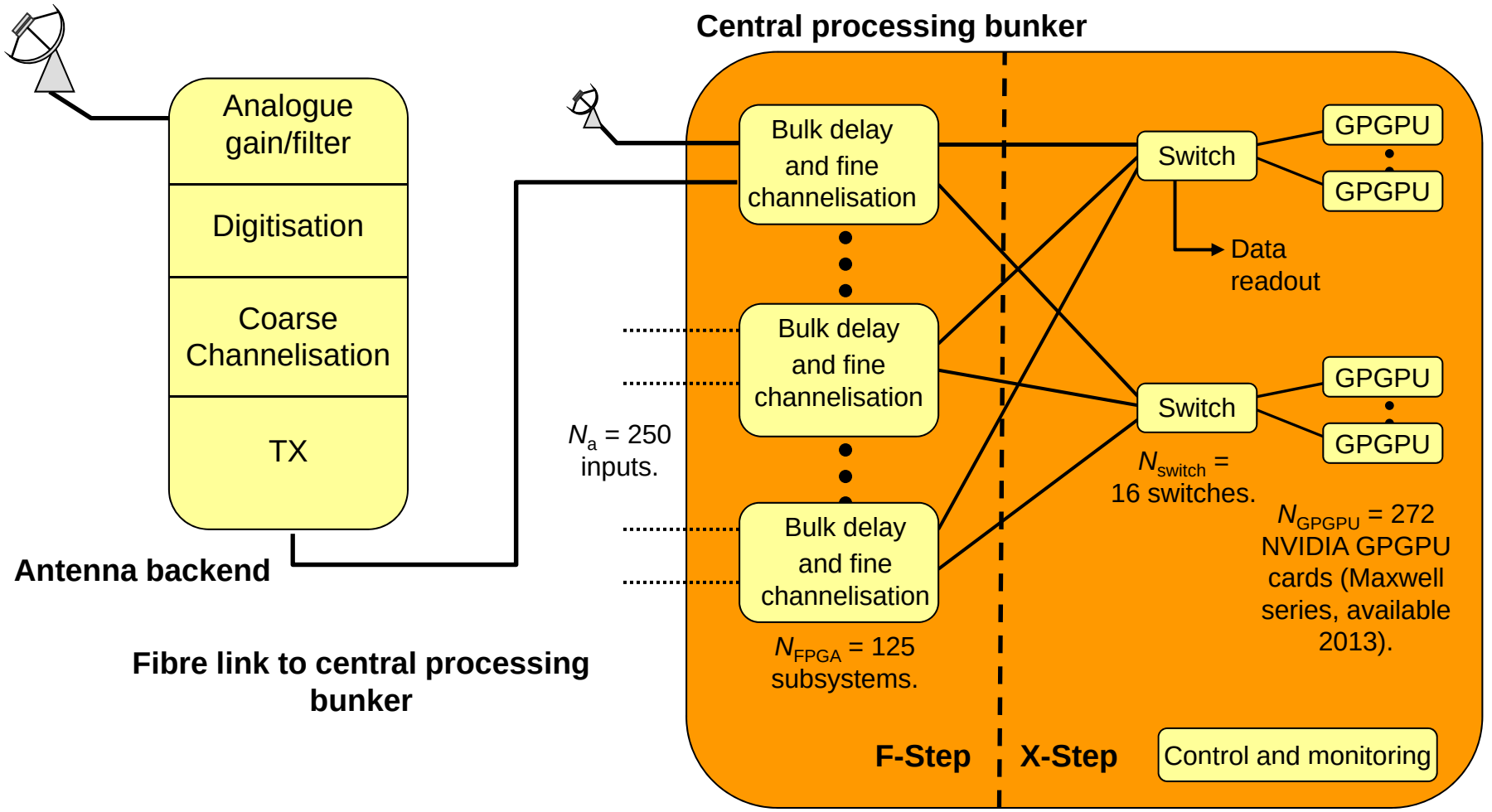
In addition, implementations of FFTs on GPUs have not yet achieved high efficiencies.

## X-step

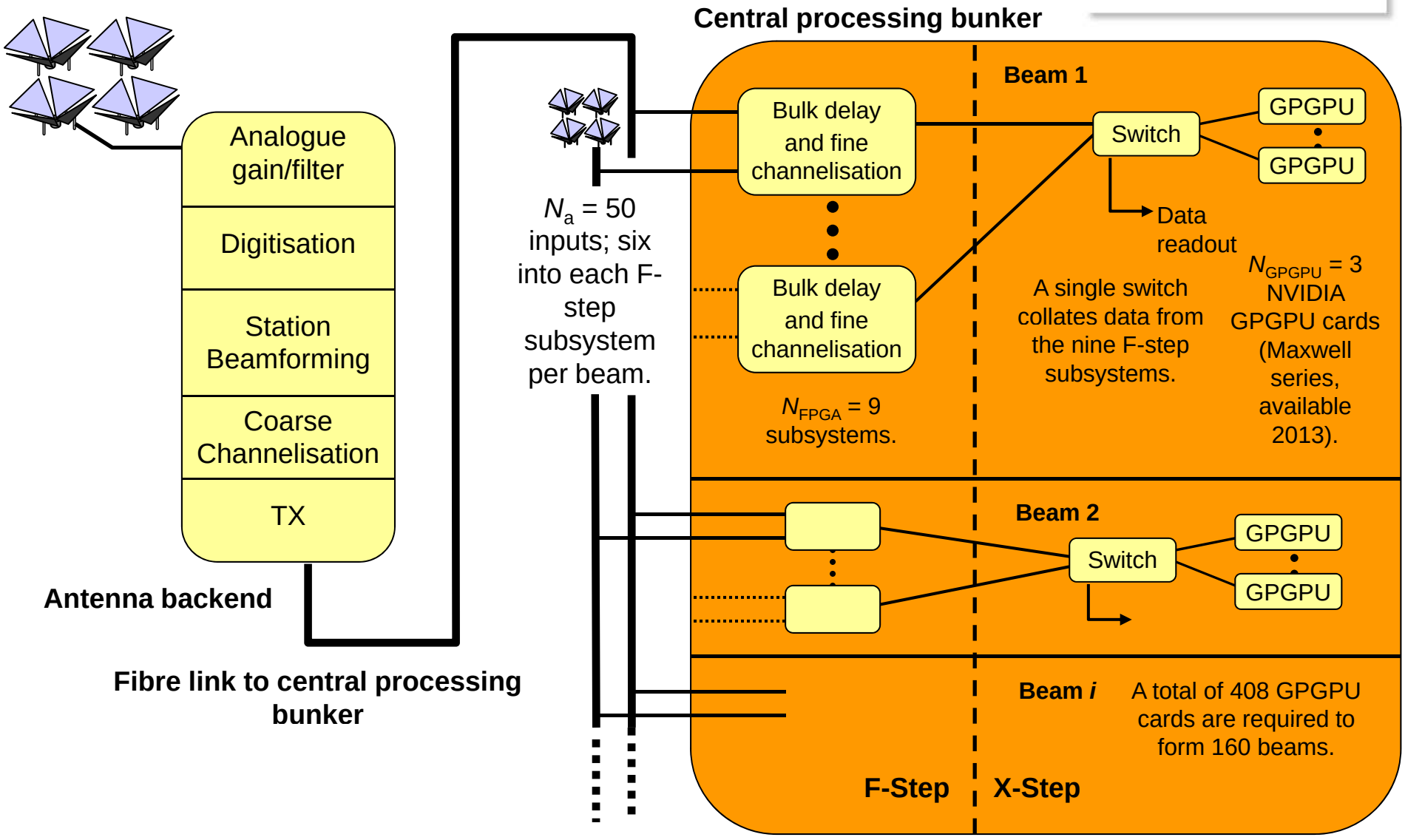
$g_{\text{X-FLOP}}/g_{\text{X-in}} = 4\, N_p N_B / N_a N_b$

This is > 16 FLOP/bit if there Are > 16 antennas.

# Software Correlator for $SKA_1$ Mid



# Software Correlator for SKA$_1$ Low



# Cost estimate (160 AA beams)

## X-step

0.17 PFLOP/s (AA)
0.002 PFLOP/s (Dishes)

## Data flow

49 Tbit/s (AA)
6.2 Tbit/s (Dishes)

## F-step

2.5 PFLOP/s (AA)
1.6 PFLOP/s (Dishes)

# Cost estimate (160 AA beams)

## F-step

0.17 PFLOP/s (AA)
0.002 PFLOP/s (Dishes)

Using ROACH boards in 2011:

1,565 boards
€7.8m

## Data flow

49 Tbit/s (AA)
6.2 Tbit/s (Dishes)

To buy using infiniband in 2011:

€0.304m (Dishes)
€0.104m (AA)

## X-step

2.5 PFLOP/s (AA)
1.6 PFLOP/s (Dishes)

240 Tesla cards (2017)

Cost: €0.6m
Power: 60kW

**Total < €9m**

# Cost estimate (480 AA beams)

## F-step

0.51 PFLOP/s (AA)
0.002 PFLOP/s (Dishes)

Using ROACH boards in 2011:

4445 boards
€22m

## Data flow

146 Tbit/s (AA)
6.2 Tbit/s (Dishes)

To buy using infiniband in 2011:

€0.912m (Dishes)
€0.104m (AA)

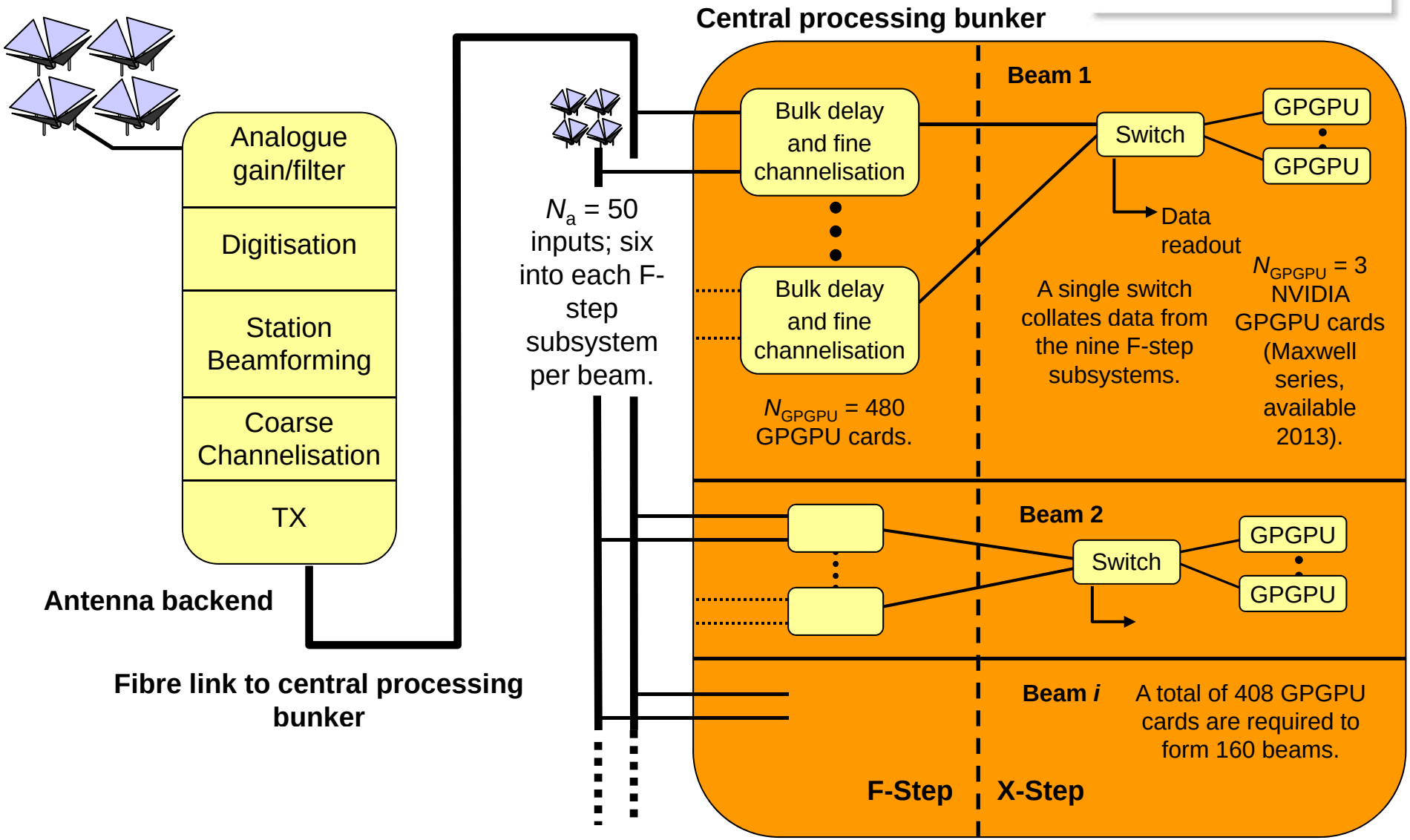
## X-step

7.4 PFLOP/s (AA)
1.6 PFLOP/s (Dishes)

560 Tesla cards (2017)

Cost: €1.4m
Power: 140kW

# Alternative Correlator for SKA$_1$ Low



# Cost estimate (480 AA beams)

## F-step

0.51 PFLOP/s (AA)
0.002 PFLOP/s (Dishes)

560 Tesla cards (2017)

Cost: €1.4m
Power: 140kW

(assuming 5% efficiency)

## Data flow

146 Tbit/s (AA)
6.2 Tbit/s (Dishes)

To buy using infiniband in 2011:

€0.912m (Dishes)
€0.104m (AA)

## X-step

7.4 PFLOP/s (AA)
1.6 PFLOP/s (Dishes)

560 Tesla cards (2017)

Cost: €1.4m
Power: 140kW

(assuming 75% efficiency)

Estimated NRE: 5-10 man-years.

# Conclusions and Future Work


- Software running on commodity processors provide a highly **flexible** way of implementing a correlator.
- The **development cost** is **low** as commodity components are used.
- In 2017, we envisage the hardware for a correlator for $SKA_1$ would cost < €4m, and dissipate around 300 kW.
- Since each GPU card correlates a fraction of the total bandwidth of $SKA_1$, the correlator can be deployed in phases.
- Hardware can potentially be shared with other applications, e.g. the beamformer or UV processor.
- There is significant existing expertise within the SKA community, including ASTRON (Romein et al.), Harvard/MWA (Greenhill et al.) and at Cambridge (Ford et al.).

# The Way Forward

- We plan to design and build a demonstrator system, which will use a small number of Tesla cards to correlate a fraction of the bandwidth of $SKA_1$.
- By evaluating its performance, we will verify (and hope to better) the efficiencies assumed here. We will also demonstrate scalability.
- We envisage that this is achievable within a year.

- ➢ We need a work package post-CoDR dedicated to investigating software correlation.
- ➢ We need a decision timescale for deployment.
- ➢ Should this work package be part of DSP or S&C?