FUTURE AND EMERGING TECHNOLOGIES
PROJECT N. 249013

SEVENTH FRAMEWORK PROGRAMME THEME
FET proactive 1 (ICT-2009.8.1)
Concurrent Tera-Device Computing

TERAFLUX, Effective Operation of Dataflow Parallelism in Teradevices

MATEO – 2012

Multicore Architectures and Their Effective Operation

Disclaimer: the statements in this presentation do not necessarily represent the position of the TERAFLUX Consortium, but only the view of the presenter.

Roberto Giorgi – University of Siena, ITALY – http://www.dii.unisi.it/~giorgi

What is **TERA^{F}LUX** about

Architecture+Programmability+Reliability

of

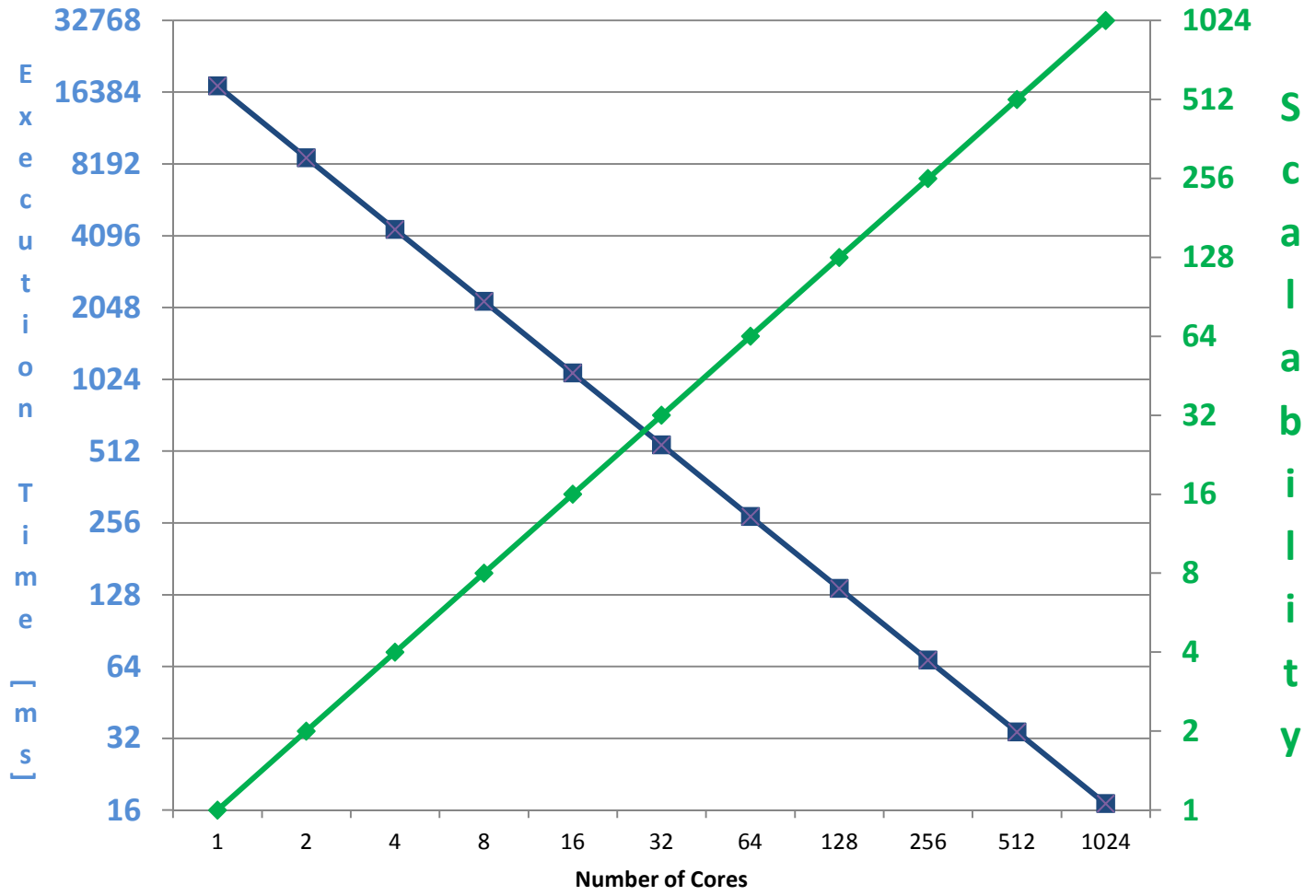
Future (single chip)

Many-cores

(targeting 1000+ cores) *

* TERADEVICE==**10^12** DEVICES
(or 1000 BILLIONS of them)

Scalability of Sequential Source Code Targeting the T* x86_64 ISA Extension



Data-Flow Parallelism Enabled this Effective Operation !

Simulation Framework

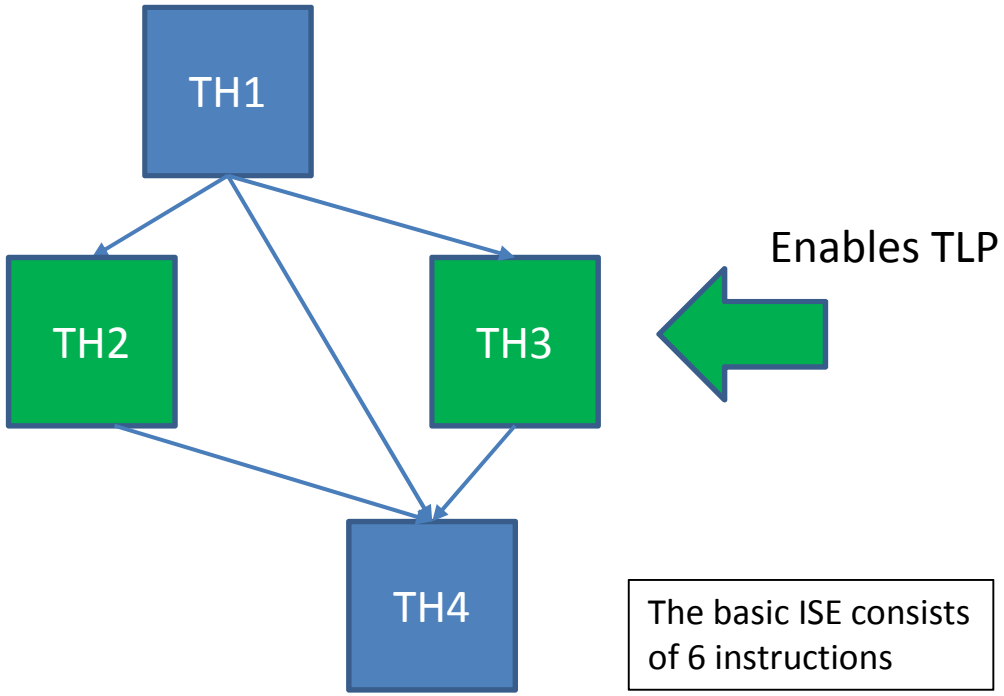
- Simulation host: 1TB, 64 x86_64 cores
- Simulation software: HPLabs COTSon
- Modeled cores: AMD SimNow cores @800MHz
- Initial pipe-cleaner: recursive Fibonacci (n=40)

What is the T* ISE ?

- A minimalistic ISA Extension of a standard ISA (e.g., x86_64 – why not?)
- The key instructions are TSCHEDULE/TDESTROY

TH1:

```
------
------
------
TSCHEDULE (&fp2, TH2, sc2)
TSCHEDULE (&fp3, TH3, sc3)
TSCHEDULE (&fp4, TH4, sc4)
------
------
TDESTROY
```

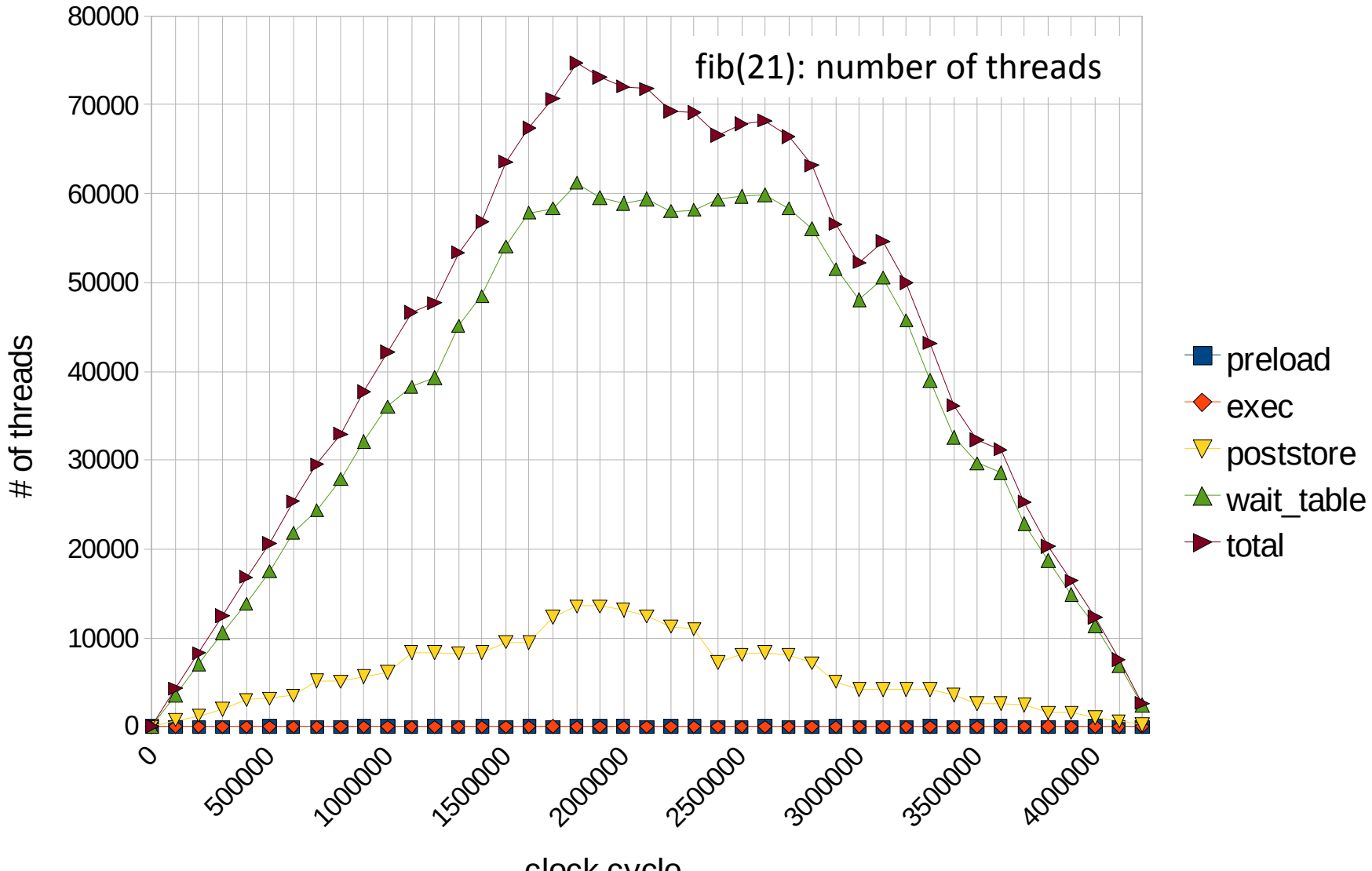


The basic ISE consists of 6 instructions

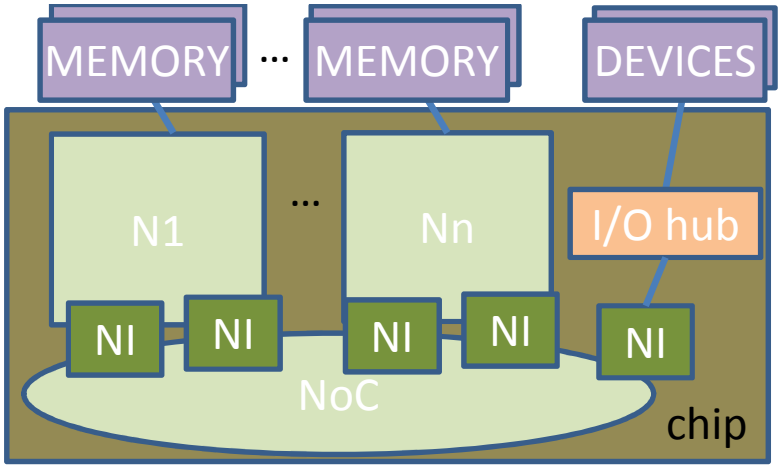
Design Goals

- Decoupling of Memory Accesses
- Asynchronous operation of Threads
- Simple and Fast ISA interface
- Scalability with the number of cores
- Non-centralized control
- All-or-none execution of the single thread
- DF Thread Isolation, Repeatability
- Suitable for Memory Transactions

Is there TLP potential?



TERAFLUX Architectural template



LEGENDA:

n = # of nodes

m = # of cores per node

u= # of DRAM controllers insisting on the Unified Physical Address Space

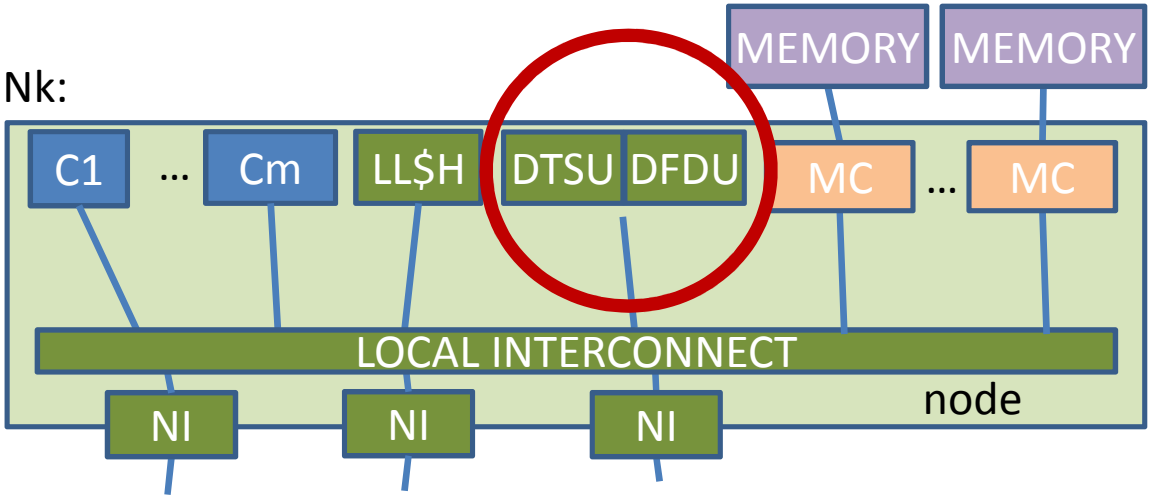
z = # of I/O Hubs

Nk = k-th Node (k=1..n)

NI = Network Interface

NoC = Network on Chip

Nk:



Cj = j-th core (j=1..m)

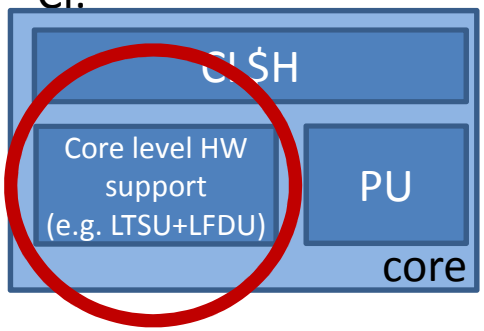
MC = Memory Controller

DTSU = Distributed Thread-Scheduler Unit

DFDU = Distributed Fault-Detection Unit

LL$H = Last Level Cache Hierarchy

Ci:

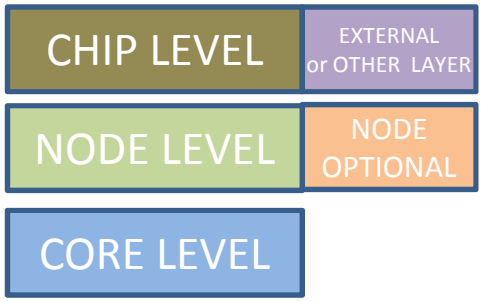


CL$H = Core Level Cache Hierarchy

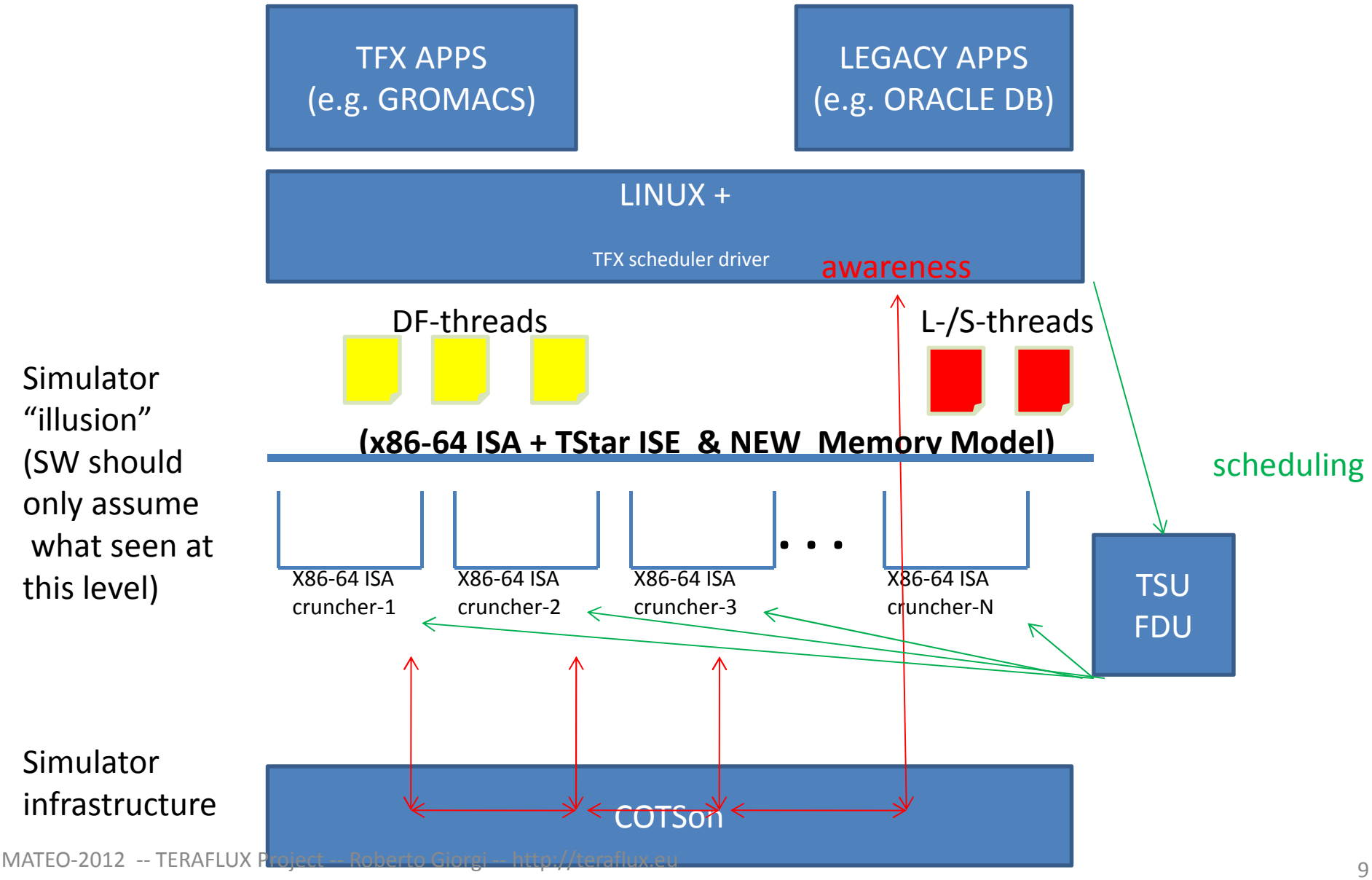
PU = Processing Unit

LTSU = Local Thread-Scheduler Unit

LFDU = Local Fault-Detection Unit



Data-Flow Threads …and… NON-Data-Flow Threads



Simulator "illusion" (SW should only assume what seen at this level)

Simulator infrastructure

Data-Flow Memory Model

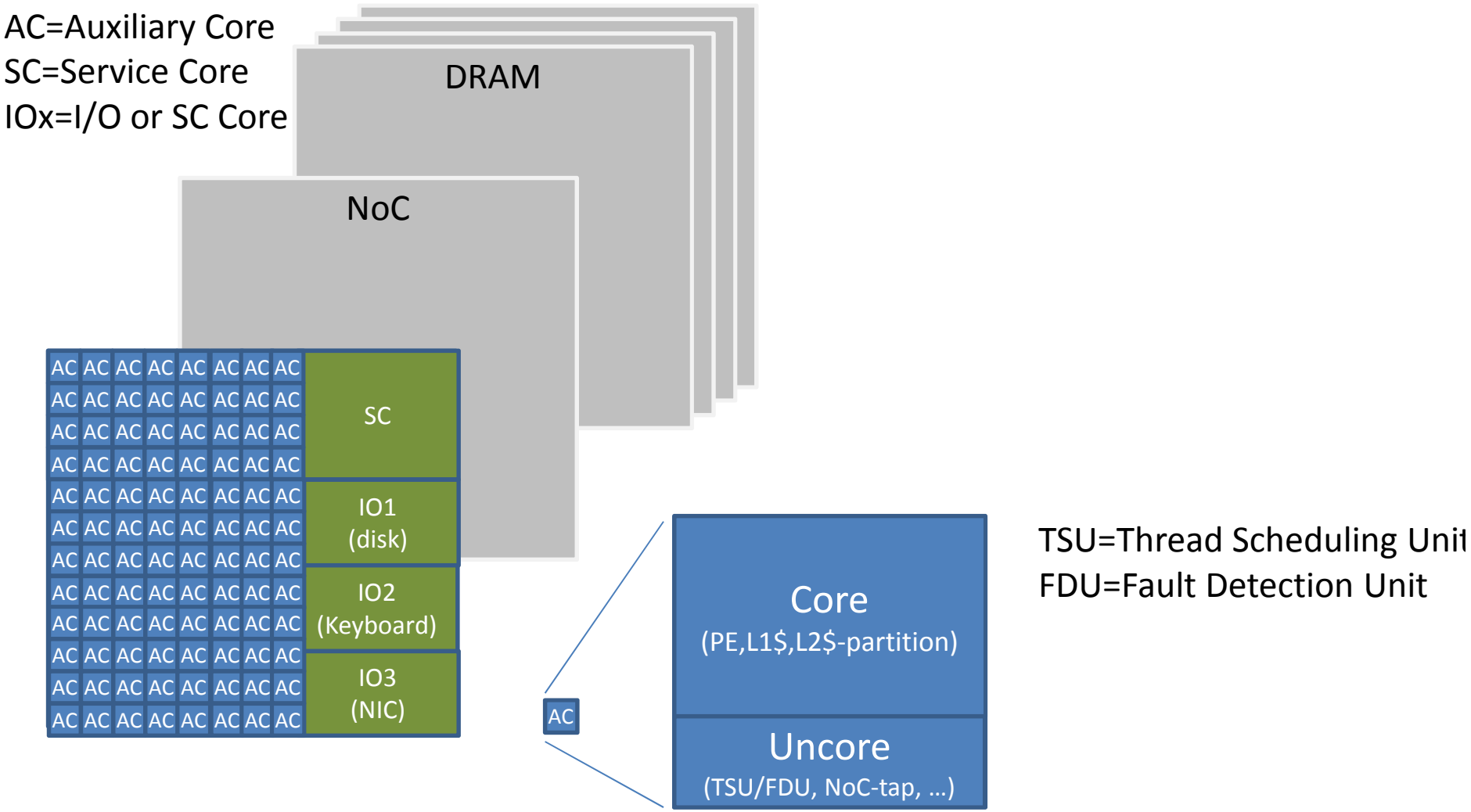
- The Software (possibly but not necessarily assisted by the hardware) should take care of **four** main producer/consumer patterns among DF threads:
  - 1-1 → private
  - N-1 → frame-memory
  - 1-N
  - N-N → shared (TM protected)

  This enables us to also get rid of hw coherency

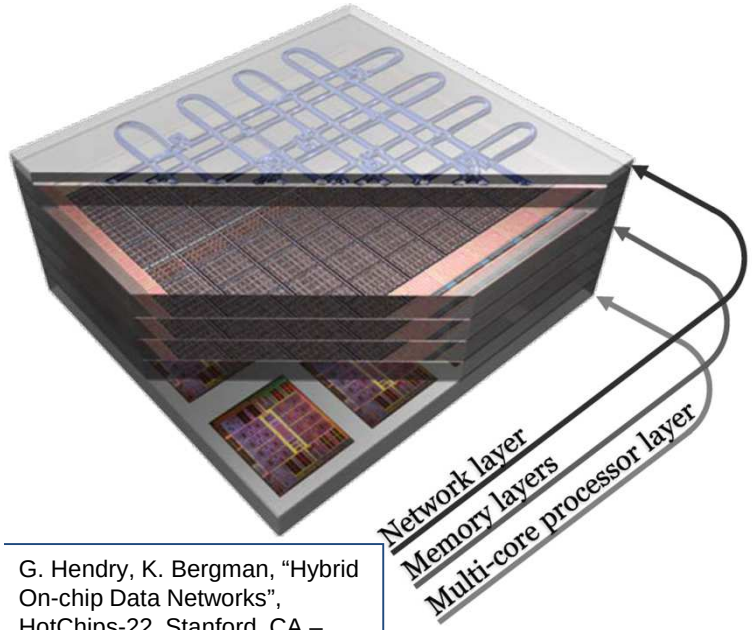
DATAFLOW

A Scheme of Computation in which an activity is initiated by presence of the data it needs to perform its function
(Jack Dennis)

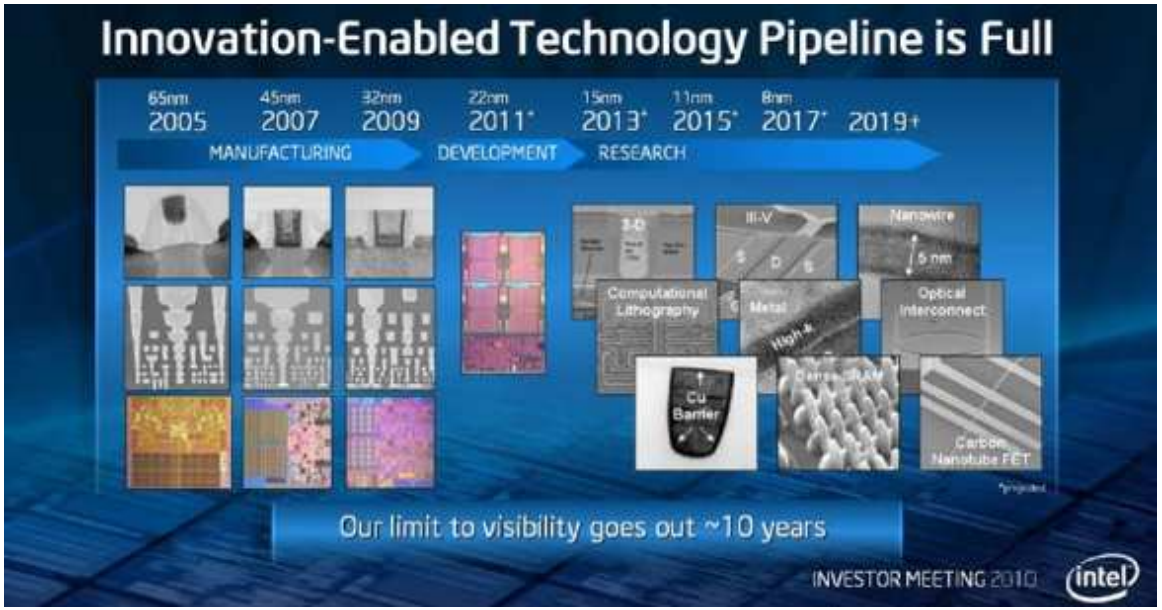
A possible TERAFLUX architectural instance



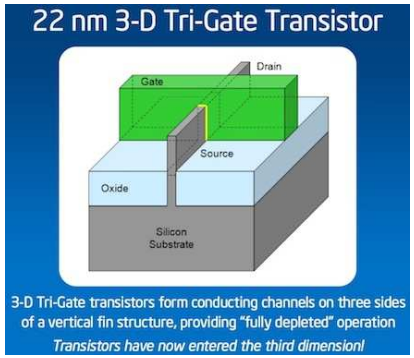
Will this fit Future Scenarios?
== 3D stacking, 8nm, 3D transistors, Graphene

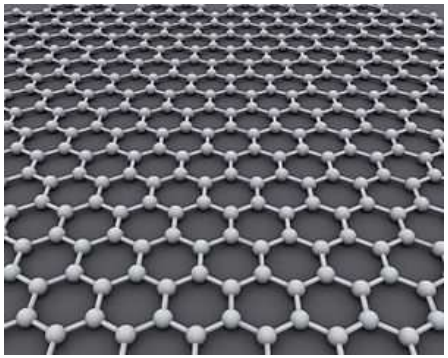


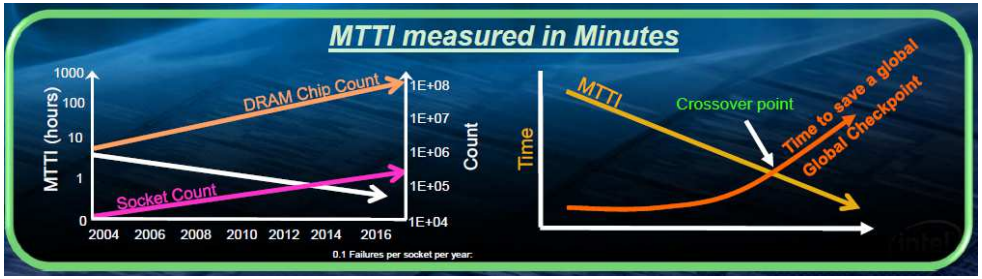
G. Hendry, K. Bergman, "Hybrid On-chip Data Networks", HotChips-22, Stanford, CA – Aug. 2010



Fab D1X (OR), 42 (AZ) starting the 14nm node in 2013







Pawloski, May 2011, Exascale Seminar, Ghent

Our pillars

- FIXED and MOST-USED ISA (**x86**)
- MANYCORE FULL SYSTEM SIMULATOR (**COTSon**)
- REAL WORLD APPLICATIONS (e.g. GROMACS)
- SYNCHRONIZATION: **TRANSACTIONAL MEMORY**
- **GCC** based TOOL-CHAIN
- OFF-THE-SHELF COMPONENTS FOR CORES, OS, NOC,MEMORY HIERARCHY
- **FDU** AND **TSU** (Fault Detection Unit and Thread Scheduling Unit)

Very inspiring work

- J. Smith – Decouple Access/Execute
- Y. Patt, W. Hwu – the Restricted Dataflow
- G. Sohi – the Multi-Scalar
- D. Burger, S. Keckler – the TRIPS architecture
- D. Culler – The TAM
- G. Gao – The Earth
- G. Papadopolus – the Monsoon

…

The Inspiration -- a.k.a. S.o.A.

- J. Dennis – for shedding the light
- I. Watson – to show that a hw was possible
- Arvind – for keep insisting on the DF potentials
- K. M. Kavi – for initiating me to the DF
- G. Gao – for showing great DF software

…

And a BIG THANK to MATEO
to give me a great courage to do this research !

University of Siena

Barcelona
Supercomputing Center

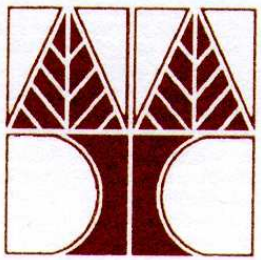
INRIA

TERA^{F}LUX.EU

Exploiting Dataflow Parallelism in Teradevice Computing

8.2 Million Euro Cost
Integrated Project
in the context of
FET program
(Future Emerging Technologies)
of the EUROPEAN UNION

University of Augsburg



University of Cyprus

University of Manchester