Spidergon a NoC for future SMP architectures

Marcello Coppola
Advanced System Technology

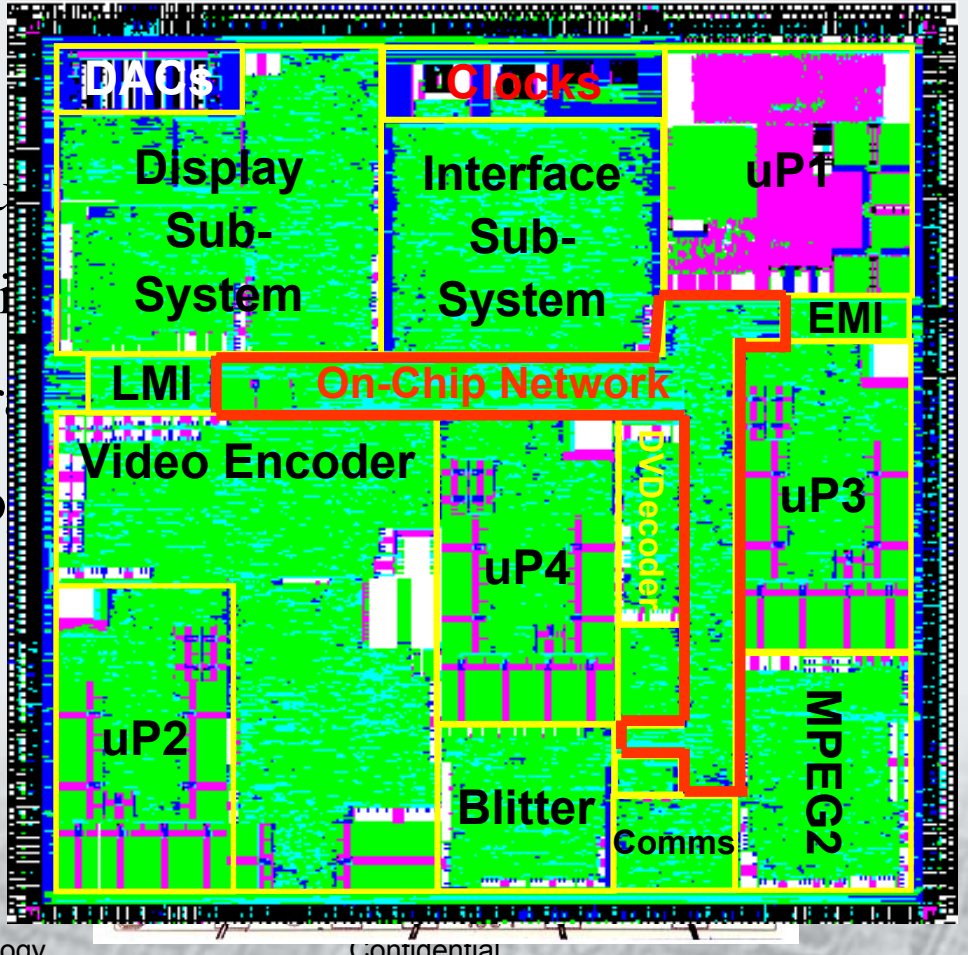
Progression of The Architecture

Vacuu

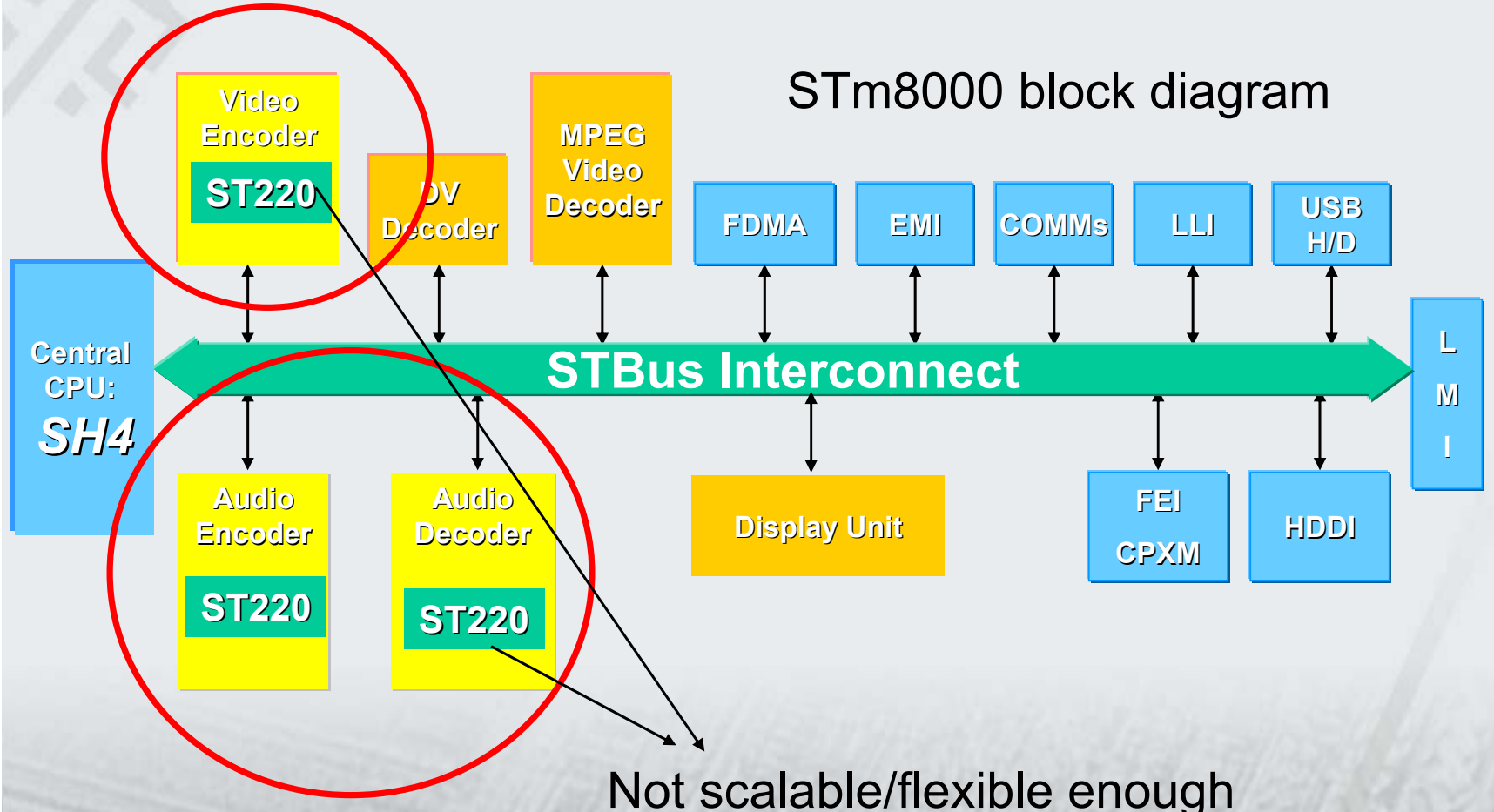
Transi

Integr

Micro



Typical Complex Multimedia Chip today:



Multi Processor SoC

- Scalability is limited by:
  - Complexity
  - Programming models limitations
  - Inadequacy of on-chip communication
  - Power consumption, both dynamic and static
- Costs issues
  - NRE as complexity increases
  - SW complexity and verification
  - HW verification time and costs

New challenging factors

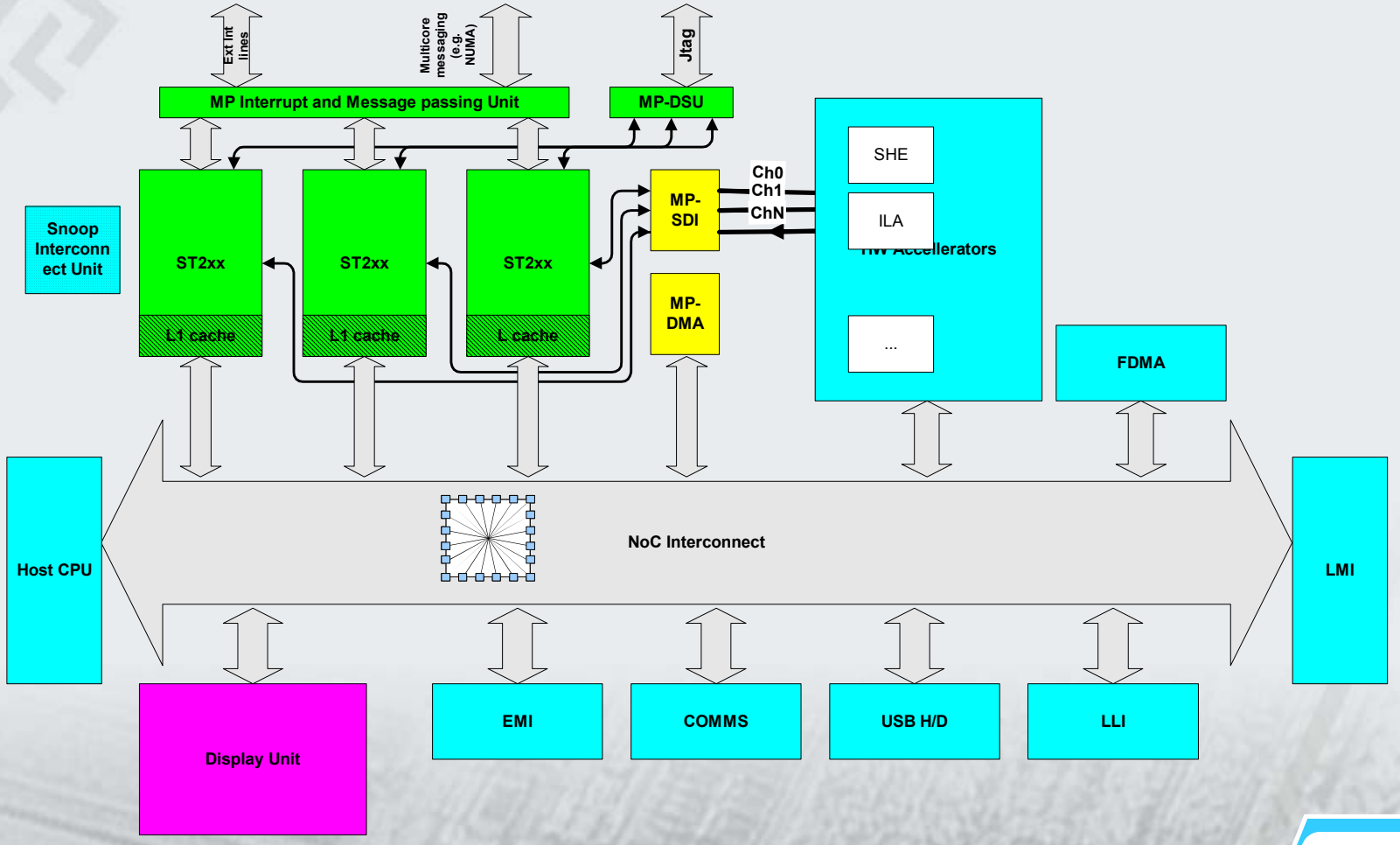
- Embedded S/W content growing fast (very fast…)
  - Technological reasons
    - ✓ H/W NRE costs exponential growth (mask, design, etc.) => ASSP's
    - ✓ Design productivity gap, time to market, flexibility
  - Growing services which require programmability
- Current application S/W complexity
  - Some internal examples
    - ✓ Set-top box: >1 million lines of code
    - ✓ Digital audio processing: >1 million lines of code
    - ✓ Recordable DVD: Over 100 person-years effort
    - ✓ Hard-disk drive: eS/W represents 100 person-years effort
  - Multi-media mobile platforms, likely worse…

⇨ Programming Model for embedded platforms is key
  - Productivity for providers as well as for their clients

Solutions

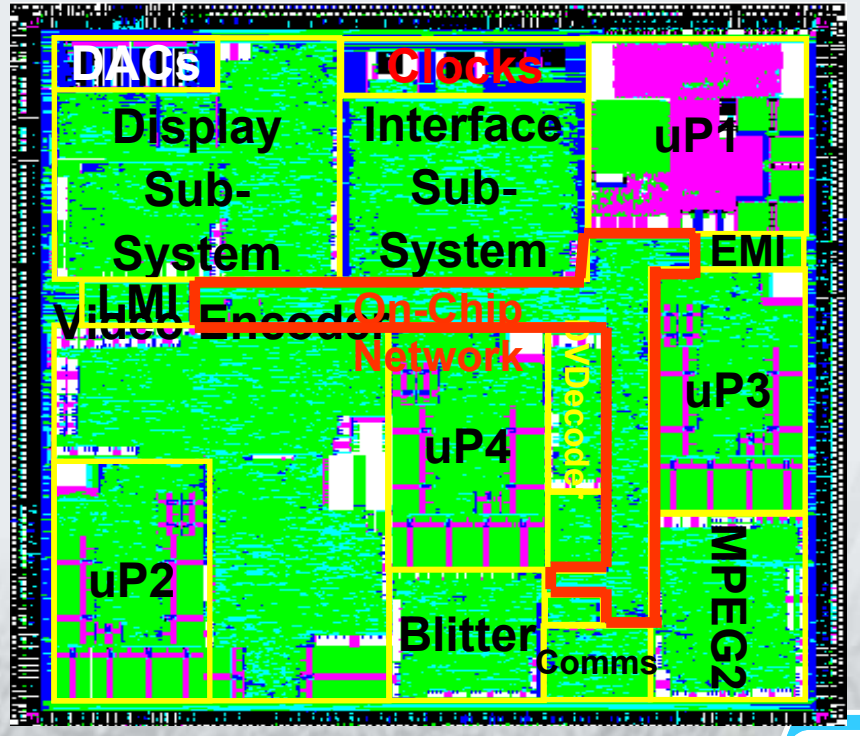
- Building multiprocessor SoC with highly symmetric IP blocks
  - Still accommodating heterogeneous system
  - Support for both symmetric multiprocessing and classical approaches
  - Easier to design and verify separately and then integrate with a well defined system template
- Flexible interconnect fabrics
  - Introduction of Noc for SoC wide communication
  - Local ‘bus like’ communication structures can still be used when required

Scalable MPSoc based on a NoC



A NoC Motivating example: STm8000 recordable dvd chip

- Stbus based,more than 40 bus targets/initiators
- Large chip area
- cross-talks problems
- clock skew



Spidergon: main goals

NoC ARCHITECTURE DEFINITION

- To be **low cost** as main drive-factor
- To differentiate from **literature**
- To exploit **OCTAGON** as starting point

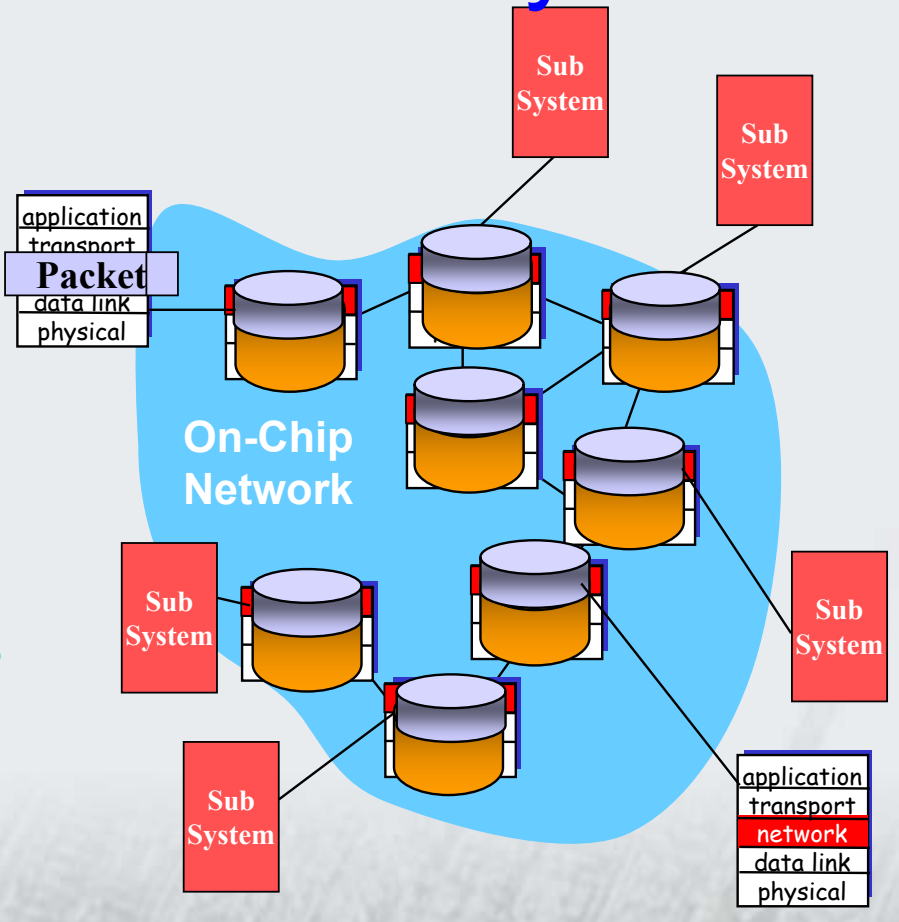
  with **STBus** experience in mind

Router up to Layer 3 - NI Layer 4

Transport packet from sending to receiving sub systems (uP,IP,etc.)

NoC is a set of

- On-chip Routers
- Network Interfaces



NoC topology: our idea

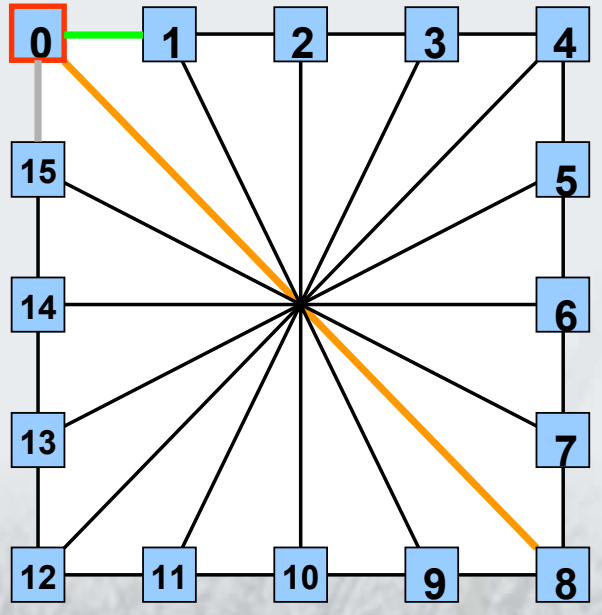
- We think that a *fixed topology* approach leads to an optimized and low cost NoC
- We want a *regular topology*
- We look for a good *complexity vs. performance* trade-off
  - Complexity: router arity, number of links, routing
  - Performance: diameter, connectivity, scalability

Spidergon topology (1/2)

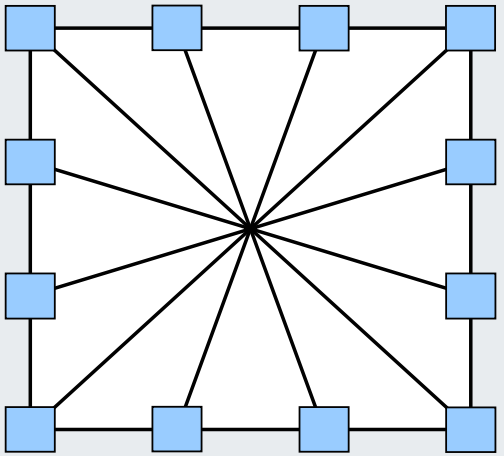
The Spidergon network connects a generic even number of nodes *N* as a bi-directional ring in both *clockwise*, and *anti-clockwise* directions with in addition a *cross* connection for each couple of nodes.

Each node *i*, with $0 \leq i < N$ is connected by two unidirectional links to:

1. node *(i+1) mod N* (*clockwise*)
2. node *(i-1) mod N* (*anti-clockwise*)
3. node *(i+N/2) mod N* (*cross*)



Spidergon topology (2/2)



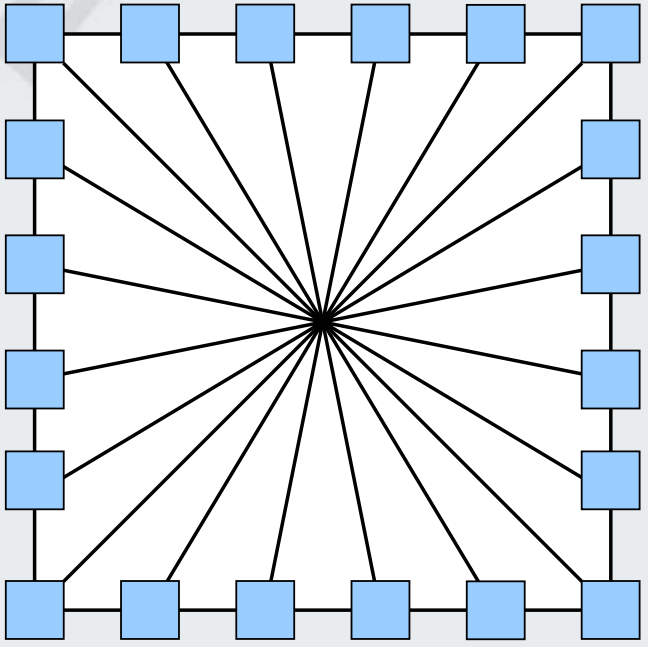
SPIDERGON TOPOLOGY PROPERTIES

Number of nodes: N = 2n with n=1,2,3,… e.g. N=14, n=7

Diameter: d = ceil(N/4) hops e.g. d=4 hops

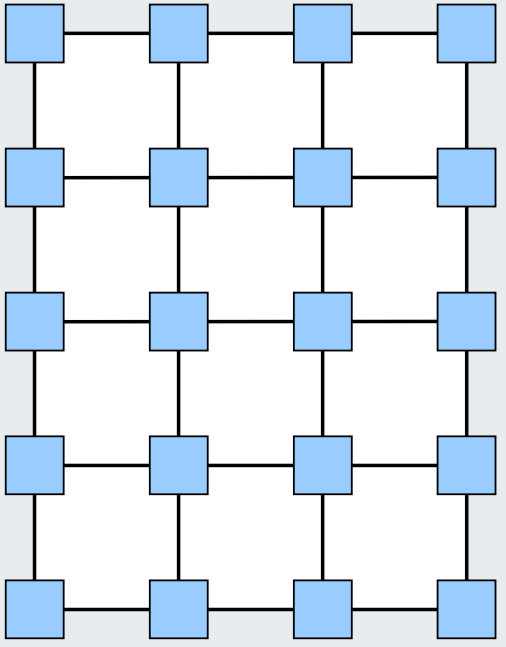
Number of links: L = 3/2 N e.g. L=21

Spidergon vs. 2D-Mesh (1/3)



N = 20 | L = 30

d_{max} = 5 hops

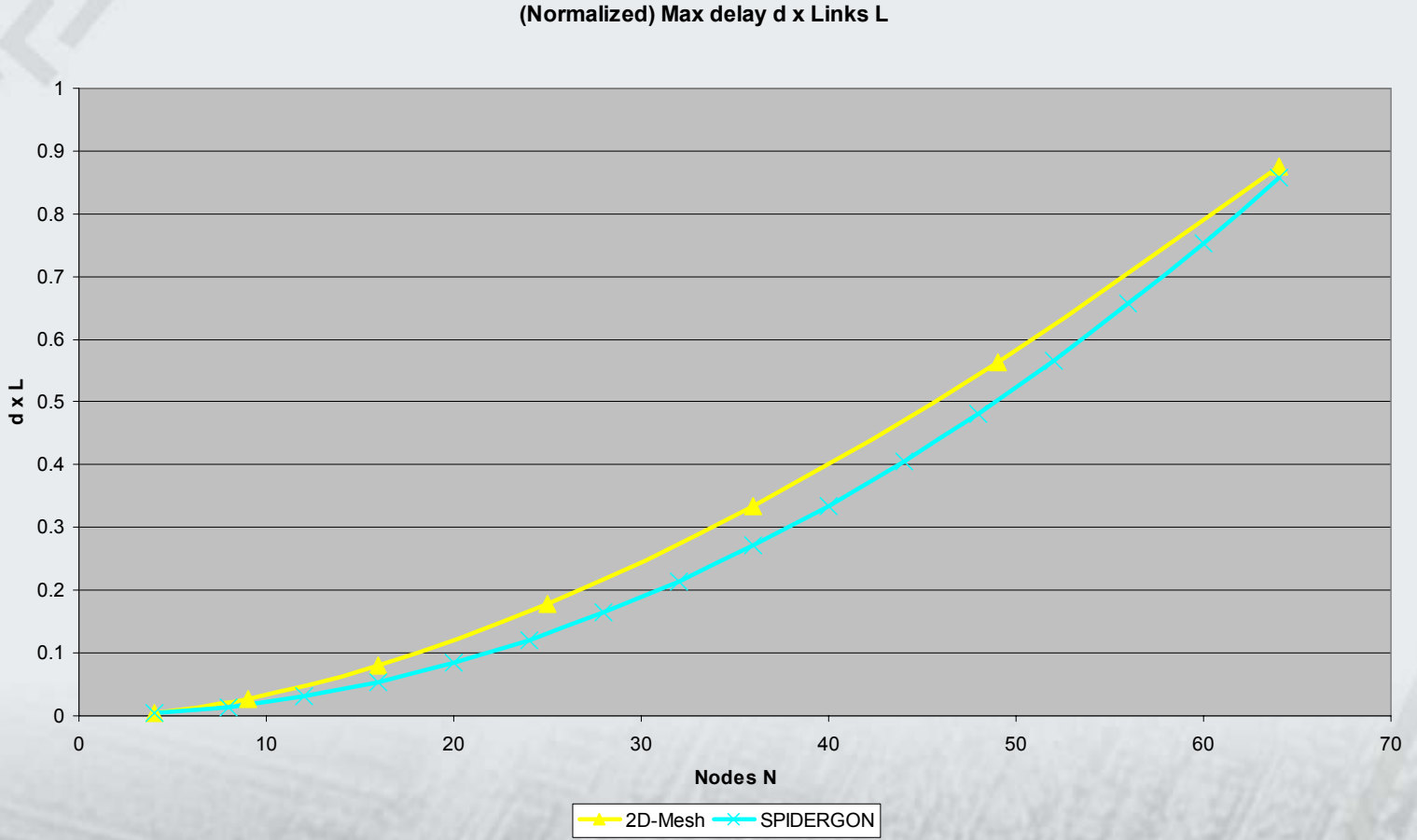


N = 20 | L = 31

d_{max} = 7 hops

Spidergon vs. 2D-Mesh (2/3)

Spidergon vs. 2D-Mesh (3/3)



Spidergon Communication: Three Levels

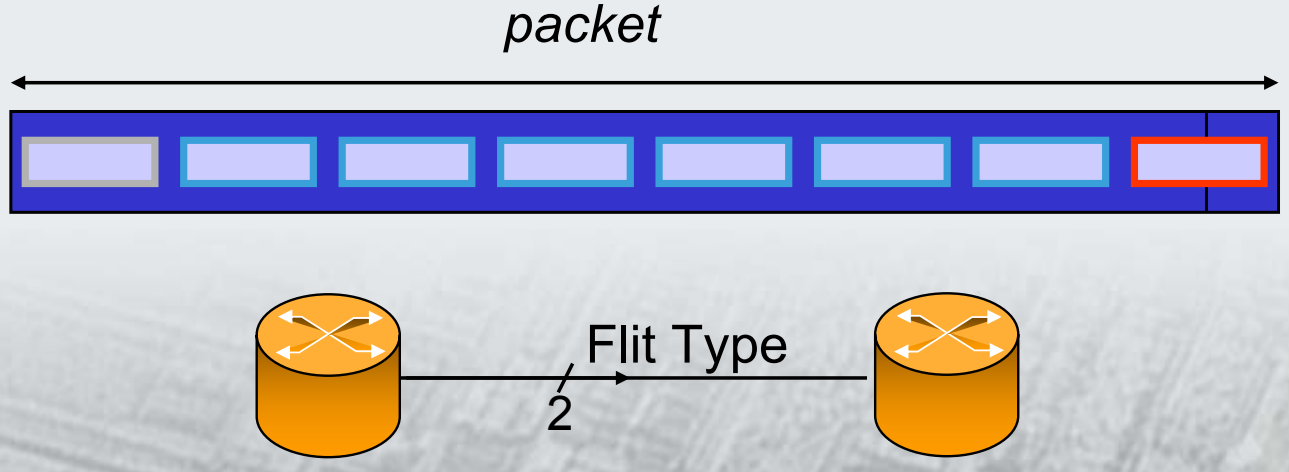
- Message
  - Unit of communication at the application level
- Packet
  - Unit of transmission
  - Each packet has a header and is routed independently
  - Large messages partitioned into multiple packets
- Flit
  - Unit of flow control
  - Each packet divided into flits that follow each other

Spidergon router design

- **Routing: virtual** wormhole (pipeline + low buffer cost)
- **Routing:** NO adaptive (ordered peer-to-peer transmission)
- **Routing:** NO table-based (area cost)
- **Queuing:** to explore the design space (output)
- **Error recovery:** NO for the moment
- **Control flow node-to-node:** very simple REQ/ACK

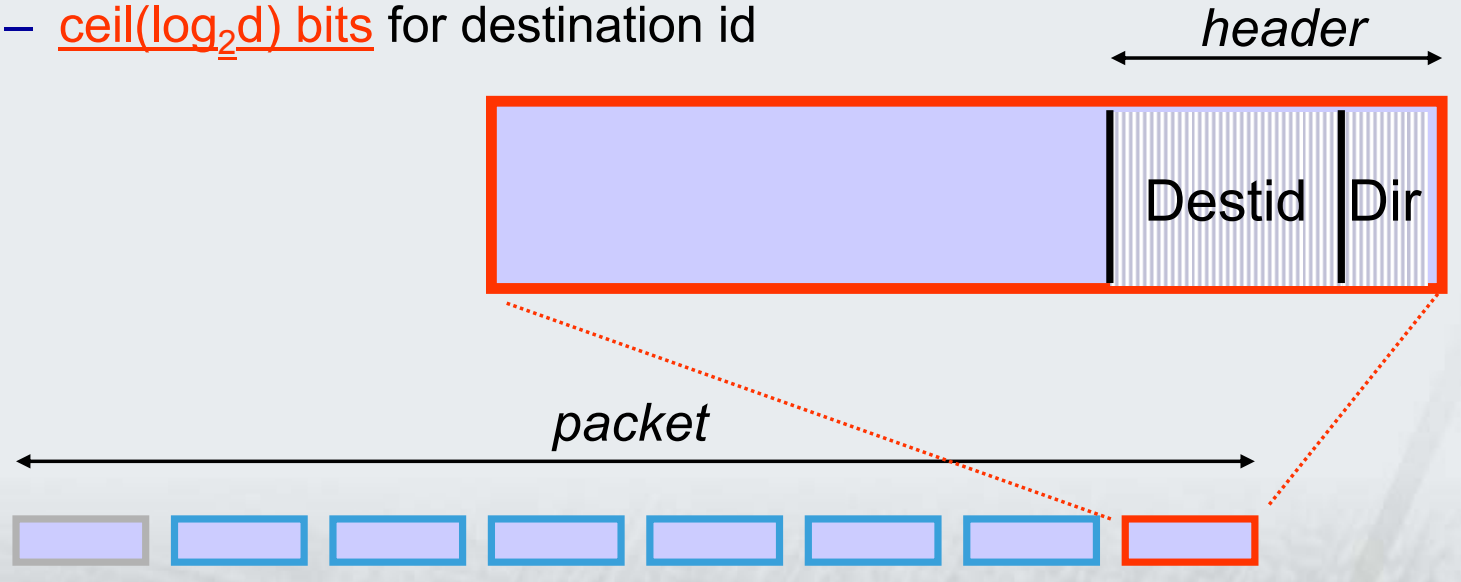
Spidergon packet

- Spidergon supports variable length packets
- A packet is composed by flits
- Flit Type: *first* – *subsequent* – *last*
  - Flit type is out of band Router-to-Router (2 bits)

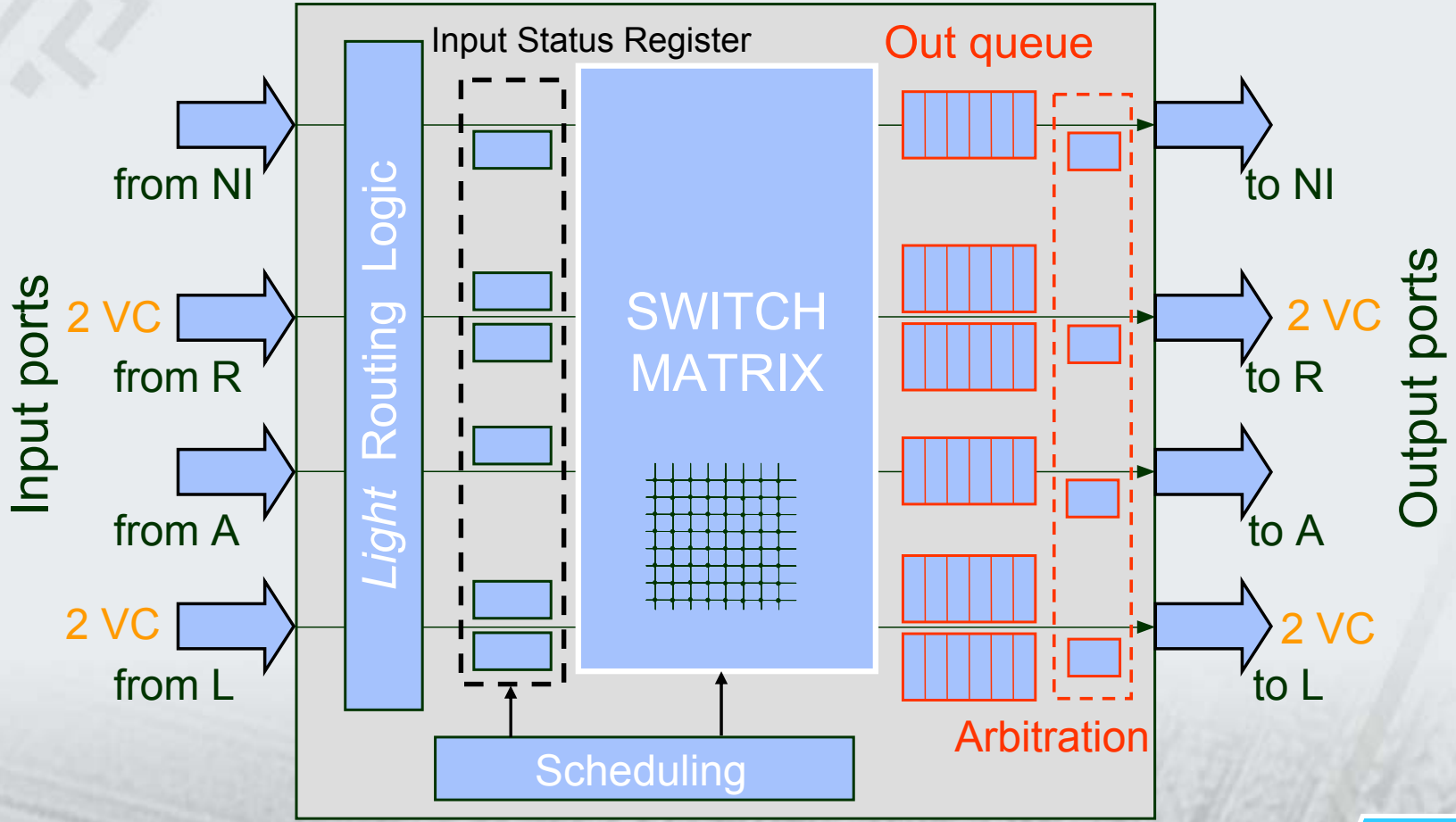


Spidergon first flit

- In the first flit (Network dependent part of the packet header):
  - 1 bit for direction
  - ceil($\log_2 d$) bits for destination id

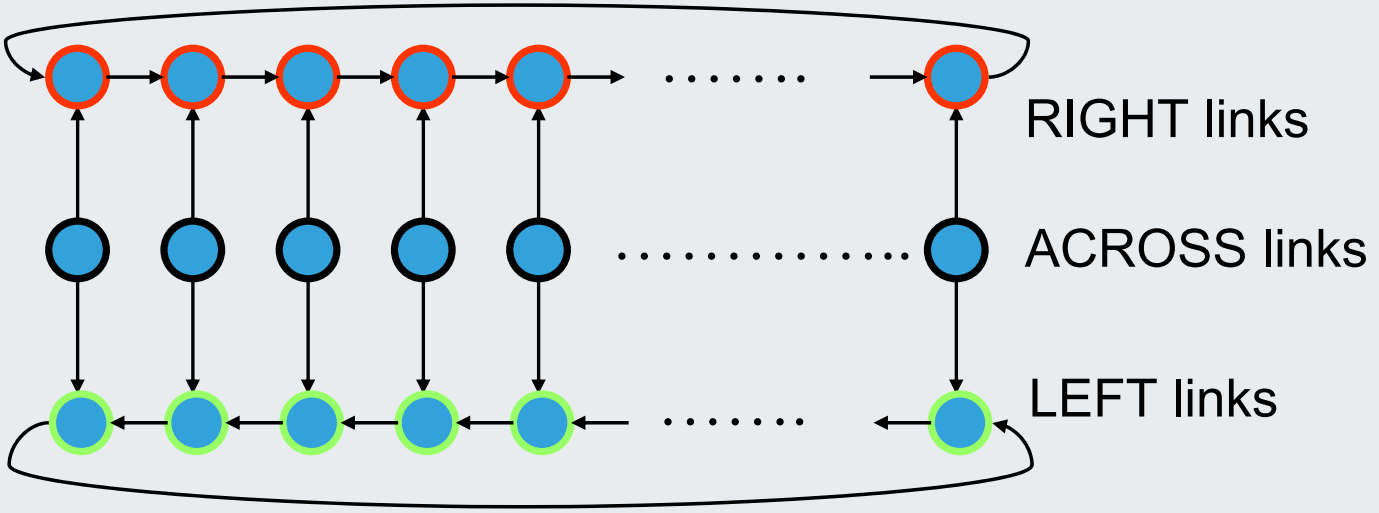


Spidergon router scheme



Deadlock issue (1/3)

Channel **D**ependency **G**raph of Spidergon



Cycles in CDG: NO DEADLOCK FREE [W.J. Dally]

Spidergon is a bidirectional ring with ACROSS links

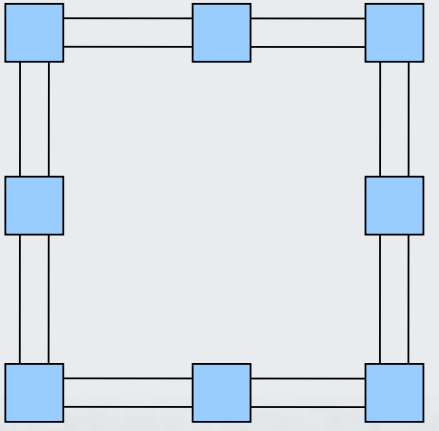
Link ACROSS used only as first hop

Deadlock issue (2/3)

To solve deadlock just consider bidirectional ring

Simple use of 2 **V**irtual **C**hannels for deadlock avoidance

(#dest - #node) $\begin{cases} > 0 \text{ then } VC_1 \\ < 0 \text{ then } VC_2 \end{cases}$



Cycles of CDG broken thanks to 2 VCs

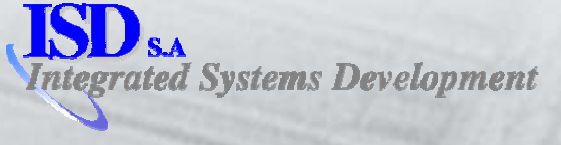
OccN partners

Open source -> http://occn.sourceforge.net

Bologna University

Pisa University

Ancona University



Network interface

- Network Interface is the “access” to the NoC
- It hides network dependent aspects
- It cover the Transport Layer

  connection handling, (dis)assembling of messages, higher level services (end-to-end protocol)
- Literature: effort to be **OCP-IP** compliant

  Open **C**ore **P**rotocol **I**nternational **P**artnership (OCP-IP)

AST target is **STBus (Type3)**

Conclusions

- AST NoC activity
- **Spidergon NoC**: a low cost architecture

  novel NoC topology

  relative routing algorithm

  simple router scheme
- **Spidergon NoC**: OCCN router model

  very flexible TLM-CA SystemC model