Address InterLeaving for Low-Cost NoCs

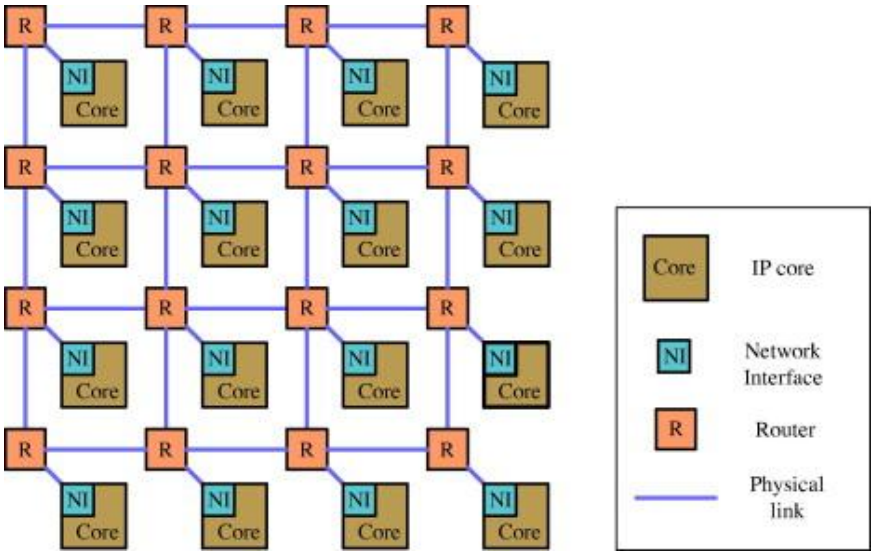
Miltos D. Grammatikakis, **Kyprianos Papadimitriou,**
Polydoros Petrakis, Marcello Coppola, and Michael Soulie

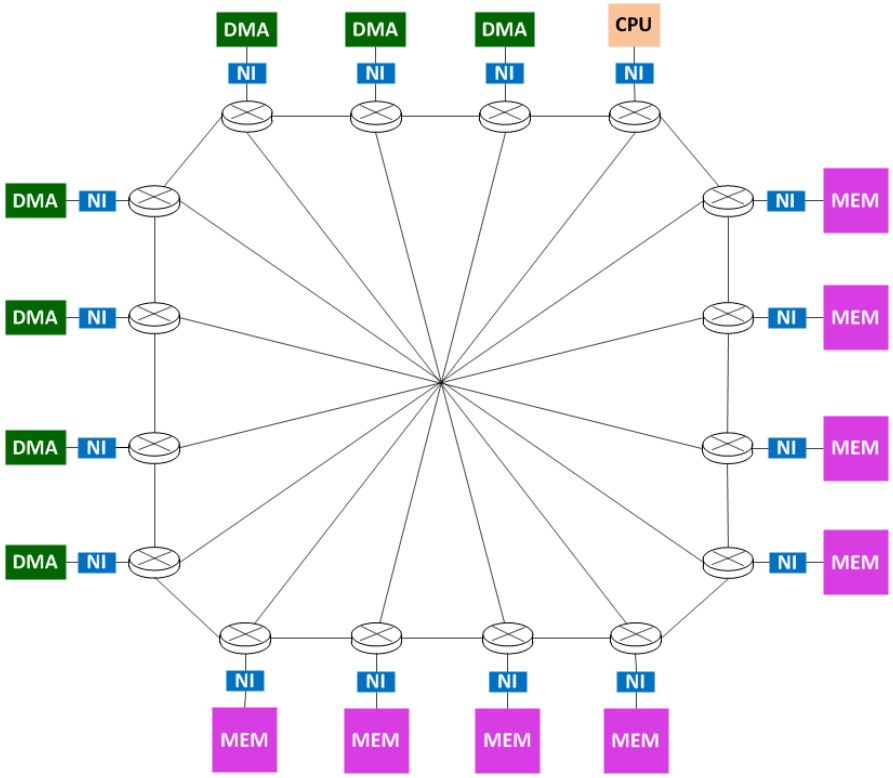
Technological Educational Institute of Crete, GR

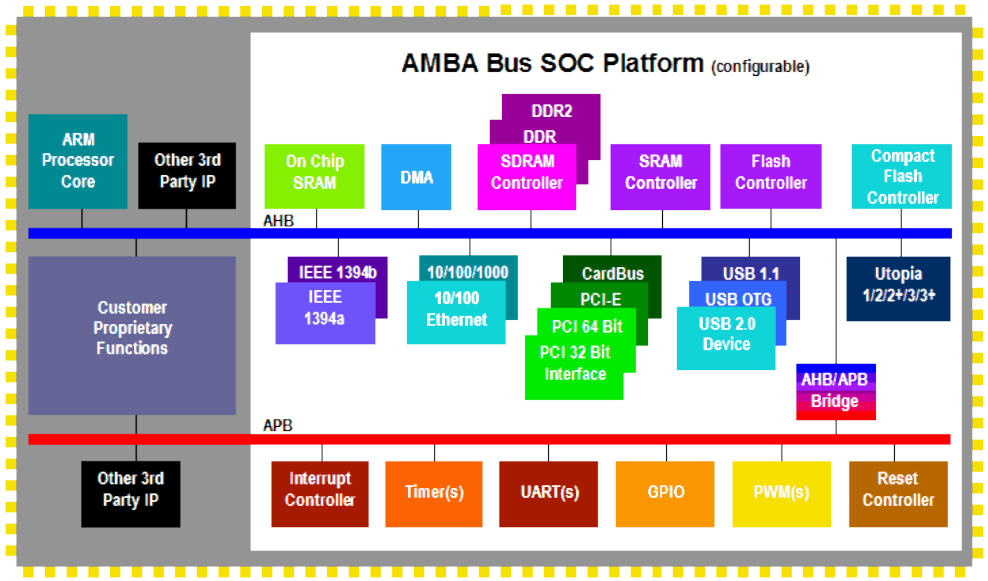
STMicroelectronics, FR

ReCoSoC Symposium, Tallinn, June 27-29, 2016

Interconnection in MPSoCs







SHAPES *EU project*

MORPHEUS *EU project*

Why Networks-on-Chip?

- Bus-based solutions don't scale
  - contention, electrical characteristics, timing, …
- Really want point-to-point links
- Full connectivity is too expensive
  - area, power, delay, …
- Ad-hoc wiring is too expensive
  - design, verification, …
- Need efficient, scalable communication fabric on chip: NoC
  - building blocks (circuits, microarchitecture)
  - topologies
  - routing & flow control schemes
  - quality-of-service (QoS)

source: D. Becker, Stanford University, 2009

The on-chip environment

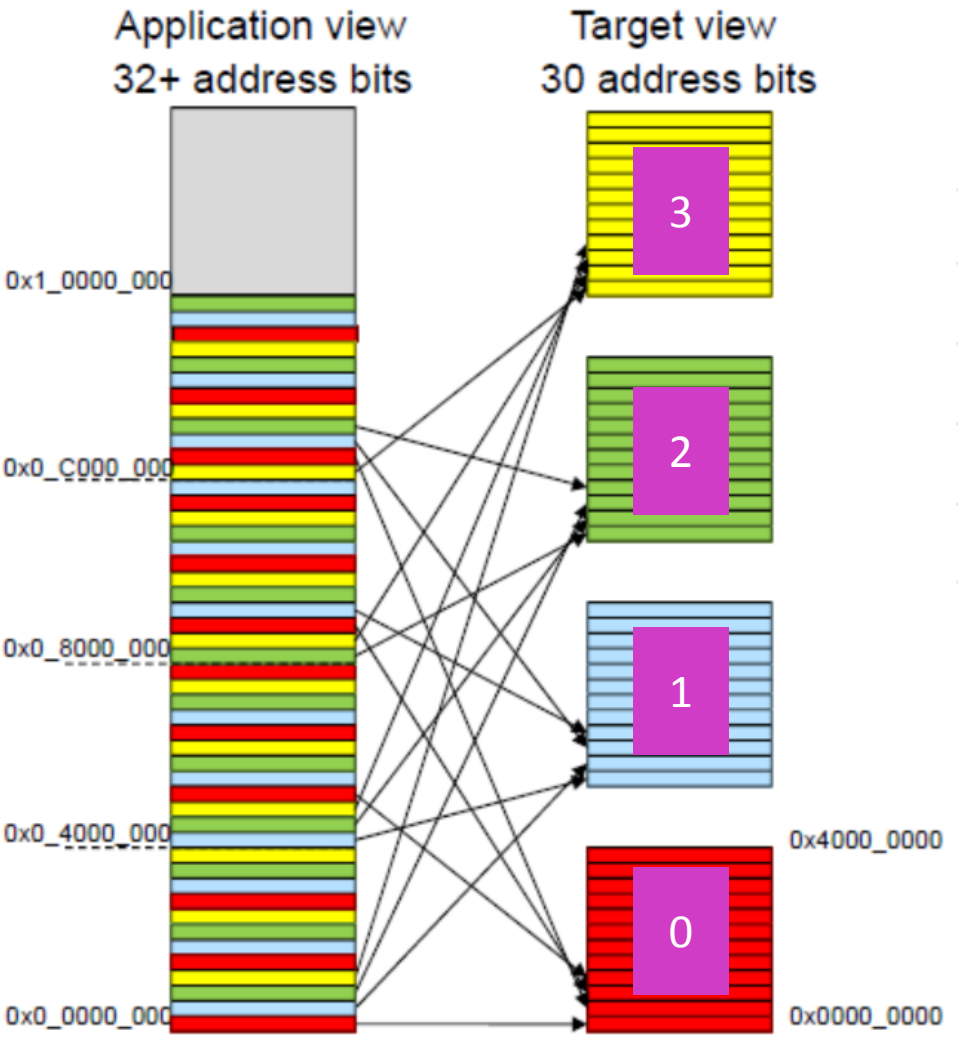
- Wires are cheap
  - favors wider interfaces and more channels, but…
- Buffers are expensive
  - provide “just enough” buffering
    - e.g. credit round trip
  - minimize occupancy & turnaround time
    - efficient flow control required
- Power budget is limiting factor for current chip designs
  - minimize power required for moving things around
  - maximize power available for doing actual work
  - NoC typically consumes > 30% of chip’s power

source: D. Becker, Stanford University, 2009

Our contribution: in NoC's QoS

- QoS-related feature
  - address Interleaving implemented at the NI
    - relies on the concept of interleaved memory
  - low-cost and low-power: with narrow links
  - estimate HW overcost
- Other QoS-related parameters
  - number of VCs
  - traffic policies
    - preemption (packet/flit interleaving)
    - fair bandwidth allocation (FBA)

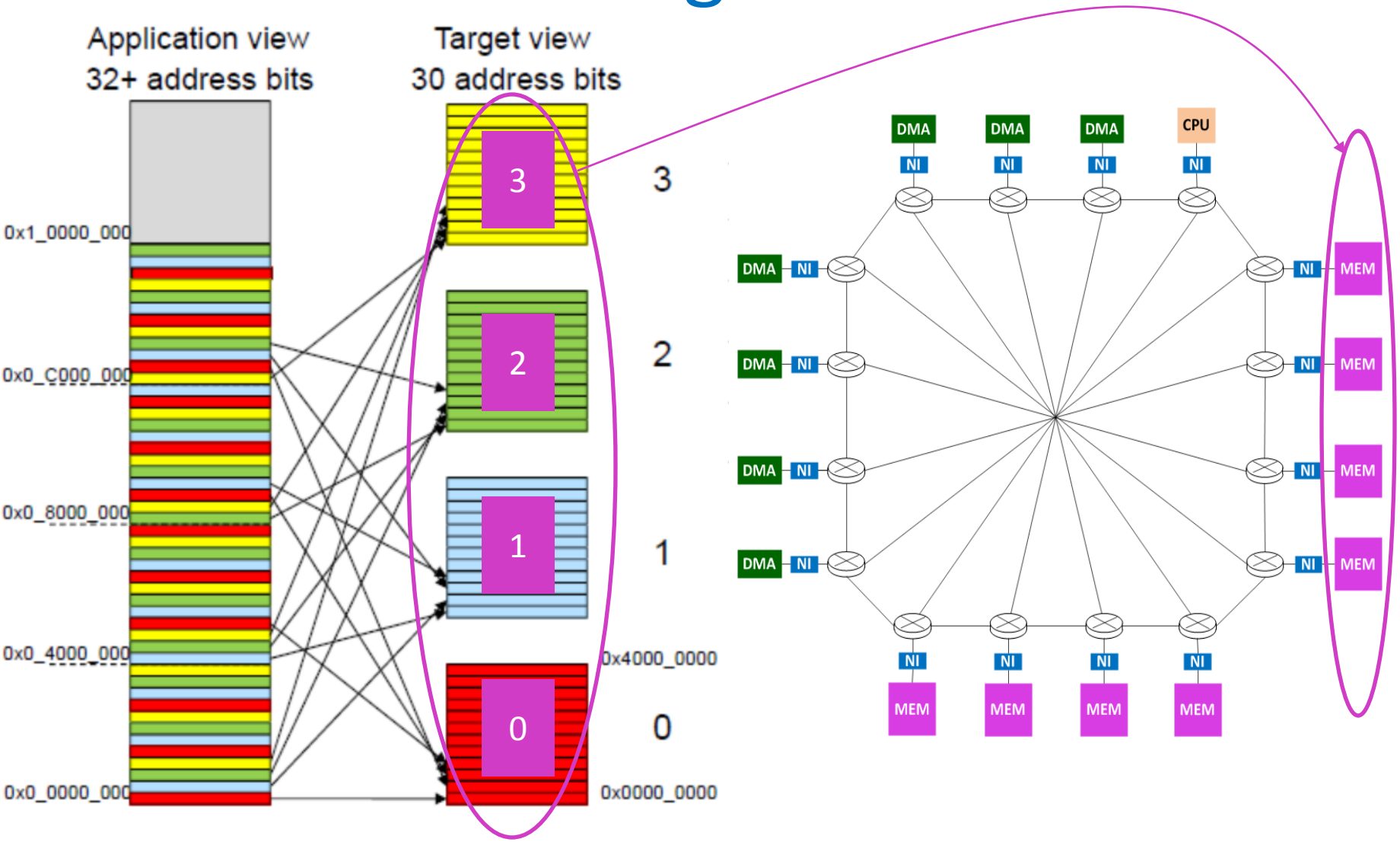
Our proposal is spreading memory..



- Spread memory space across different nodes in the MPSoC; one DDR channel on each node

- ..these nodes should be contiguous

..across contiguous nodes



*platform.inoc_system_settings (platform.inoc) | platform

iNoC

- Memory Map
  - Last Level
  - Address Interleaving
  - STNoC
- ID/Source Map
- Register Map
- Connectivity Matrix
- Coherent Connectivity Matrix
- Services

using the iNoC tool at design time..

Interleaved Memory Regions Sets Definition

Add Set | Remove Set

SetID	#NB channels	Step Size	Start @	End @	size
IMRSet_0	16	4096	0x0000_0000	0x07ff_ffff	128 MB
IMRSet_1	8	4096	0x0800_0000	0x0fff_ffff	128 MB

Interleaved Memory Regions Sets Configuration

Masters associated with IMRSet

- INI_AXI_128_RO
- INI_AXI_128_WO
- INI_AXI_32_RO
- INI_AXI_32_WO
- INI_AXI_64_RO
- INI_AXI_64_WO
- INI_STBUS_128
- INI_STBUS_32
- INI_STBUS_64

Add Master | Remove Master

IMRSet Parameters and channels configuration

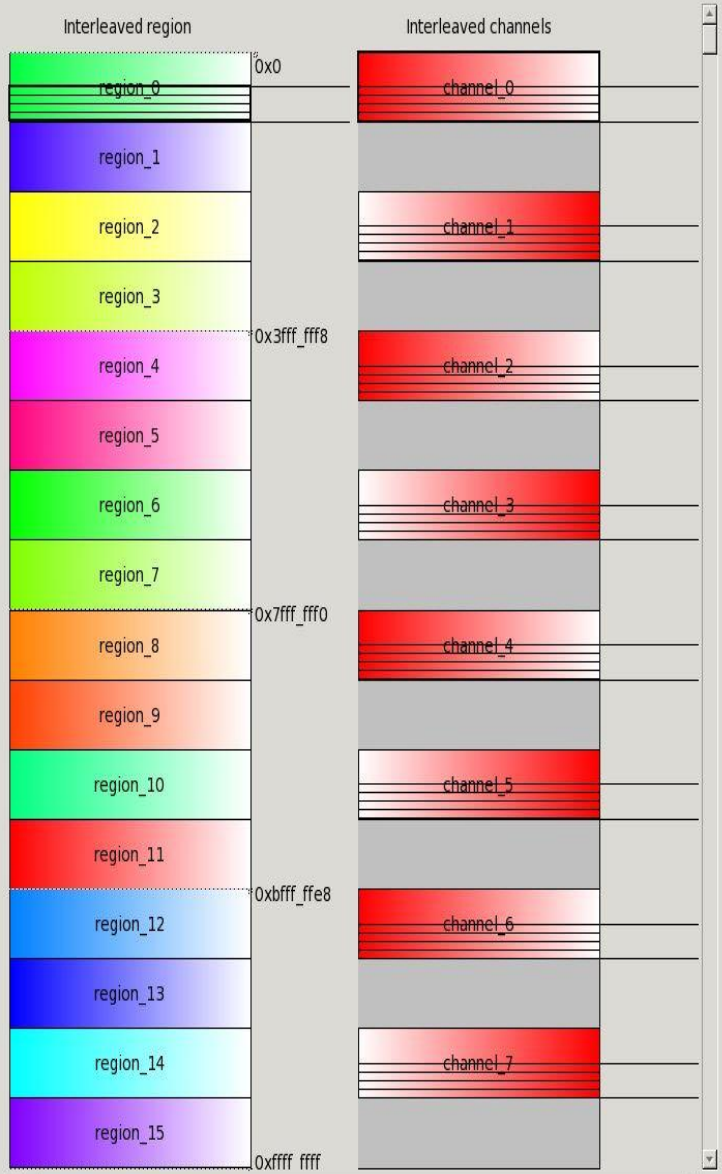
Active: true

Qos_mode: default | FBA Pri: 0 | FBA Threshold: 0

Add channel region | Remove channel region

Channel_id	Channel ranges	Reference	Default TNI
channel_0	region_0 [0000_0000 , 0fff_ffff]	true	AXI_RO_0
channel_1	region_2 [2000_0000 , 2fff_ffff]	true	AXI_RO_2
channel_2	region_4 [4000_0000 , 4fff_ffff]	true	AXI_RO_4
channel_3	region_6 [6000_0000 , 6fff_ffff]	true	AXI_RO_6
channel_4	region_8 [8000_0000 , 8fff_ffff]	true	stbus_1
channel_5	region_10 [a000_0000 , afff_ffff]	true	stbus_3
channel_6	region_12 [c000_0000 , cfff_ffff]	true	stbus_5
channel_7	region_14 [e000_0000 , efff_ffff]	true	stbus_7

Interleaved Channels View



iNoC

- Memory Map
  - Last Level
  - Address Interleaving
  - STNoC
- ID/Source Map
- Register Map
- Connectivity Matrix
- Coherent Connectivity Matrix
- Services

Default View | Memory Map width: 4 GB

Add Range | Delete

regior	lo	hi	size
region	0x0000_0000	0x0fff_ffff	256 MB
region	0x1000_0000	0x1fff_ffff	256 MB
region	0x2000_0000	0x2fff_ffff	256 MB
region	0x3000_0000	0x3fff_ffff	256 MB
region	0x4000_0000	0x4fff_ffff	256 MB
region	0x5000_0000	0x5fff_ffff	256 MB
region	0x6000_0000	0x6fff_ffff	256 MB
region	0x7000_0000	0x7fff_ffff	256 MB
region	0x8000_0000	0x8fff_ffff	256 MB
region	0x9000_0000	0x9fff_ffff	256 MB
region	0xa000_0000	0xafff_ffff	256 MB
region	0xb000_0000	0xbfff_ffff	256 MB
region	0xc000_0000	0xcfff_ffff	256 MB
region	0xd000_0000	0xdfff_ffff	256 MB
region	0xe000_0000	0xefff_ffff	256 MB

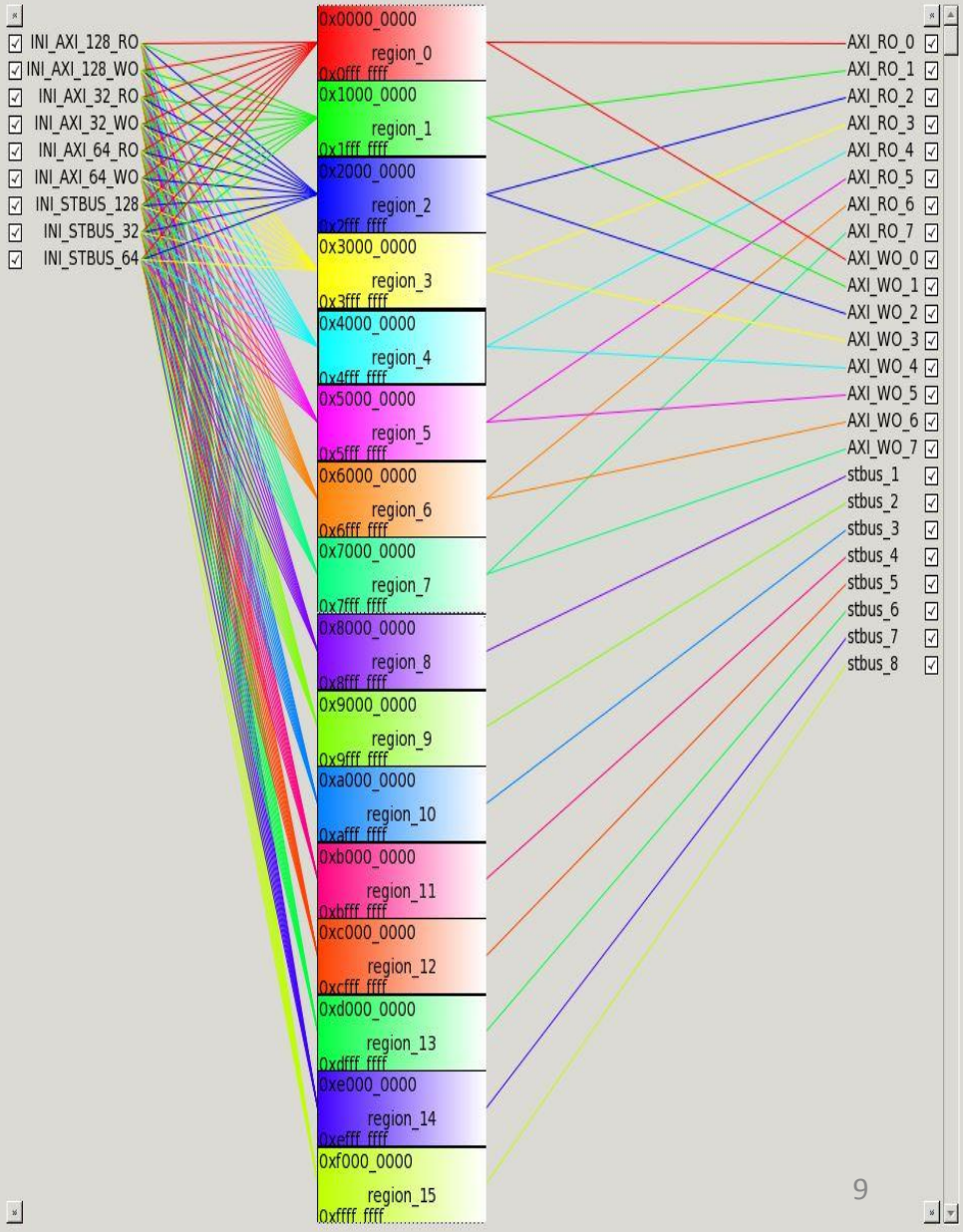
NI Initiators/Targets

Add ... | Remove ...

Default

INI_AXI_128_RO -> Undefined

INI_AXI_128_RO	AXI_RO_4
INI_AXI_128_WO	AXI_WO_4
INI_AXI_32_RO	
INI_AXI_32_WO	
INI_AXI_64_RO	
INI_AXI_64_WO	
INI_STBUS_128	
INI_STBUS_32	
INI_STBUS_64	



..for memory mapping

IMR and IMRSets

- 1 channel can have multiple Interleaved Memory Regions (IMRs)
  - each of these belongs to a different IMRSet
- IMRs belonging to different channels are grouped in 1 IMRSet
  - interleaving within 1 IMRSet <-> different channels!
- Up to 16 IMRSets can be supported
  - requires de-interleaving implementation at target NI
  - *we experimented with 1 IMRSet only*

Hardware cost

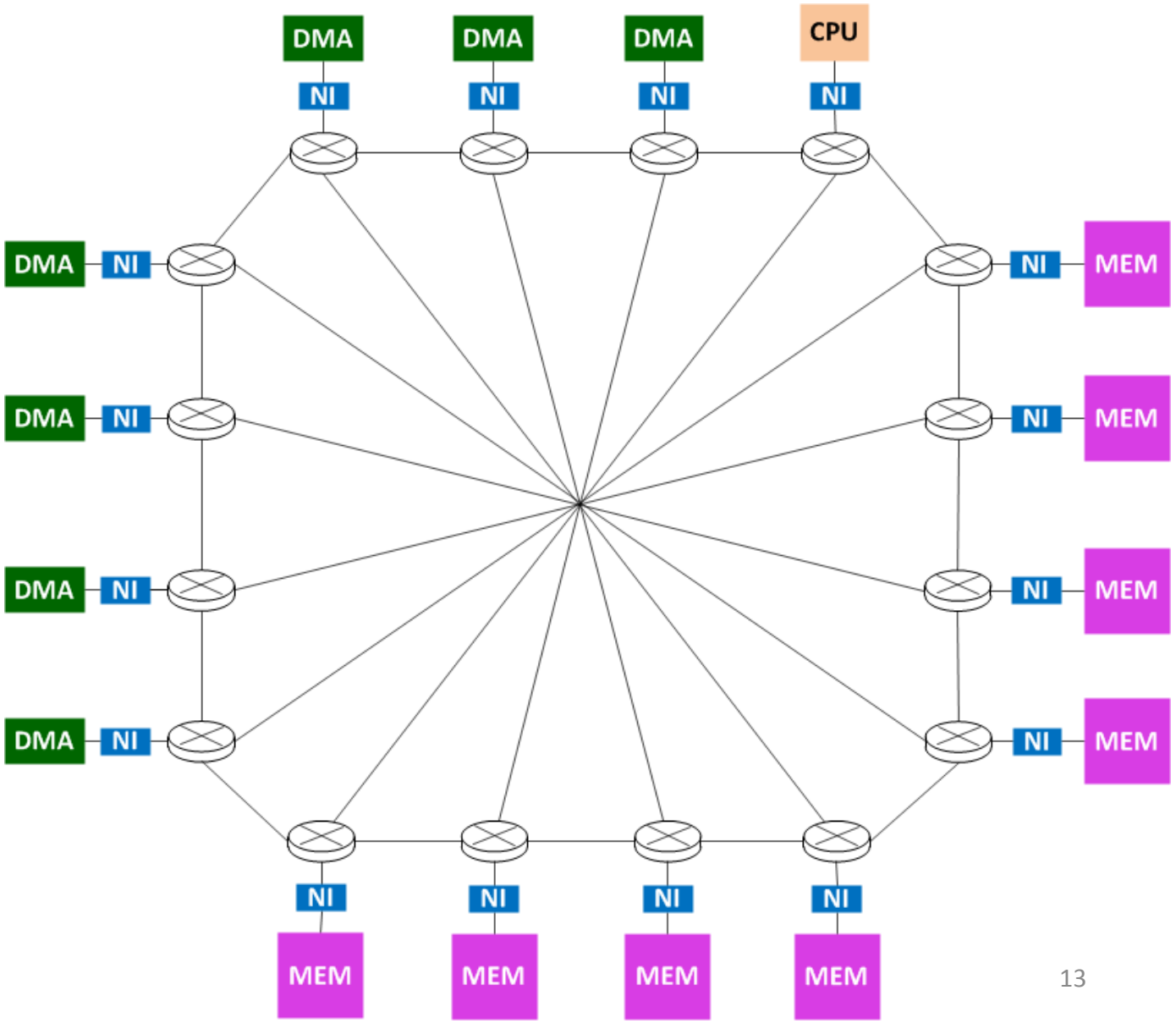
- Non-disruptive modification of STNoC routing
  - address comparators enhanced with an output signal indicating whether incoming address belongs to an IMR
- Typical configuration adds 5% more gates
  - synthesized in 28nm FDSOI ST technology

Address interleaving: then and now

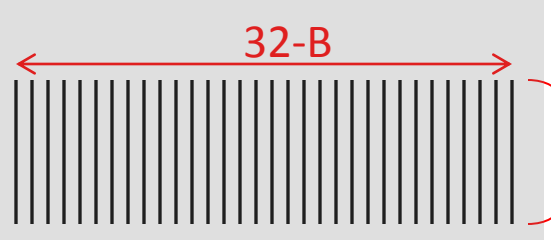
- Initially it was proposed to hide the memory refresh time of DRAMs; used in IBM 360/85, Ultrasparc III, Cray-Y MP

- Different channels to balance network load amongst them; balance also the traffic in the NoC itself
- Implemented in the initiator's network interface (INI)
- Transactions are split and interleaved within the interconnect and the NIs
- Decision to which channel a transaction belongs, depends on the transaction address

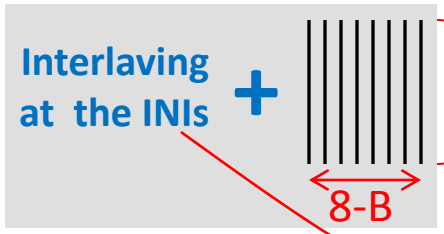
Benefits

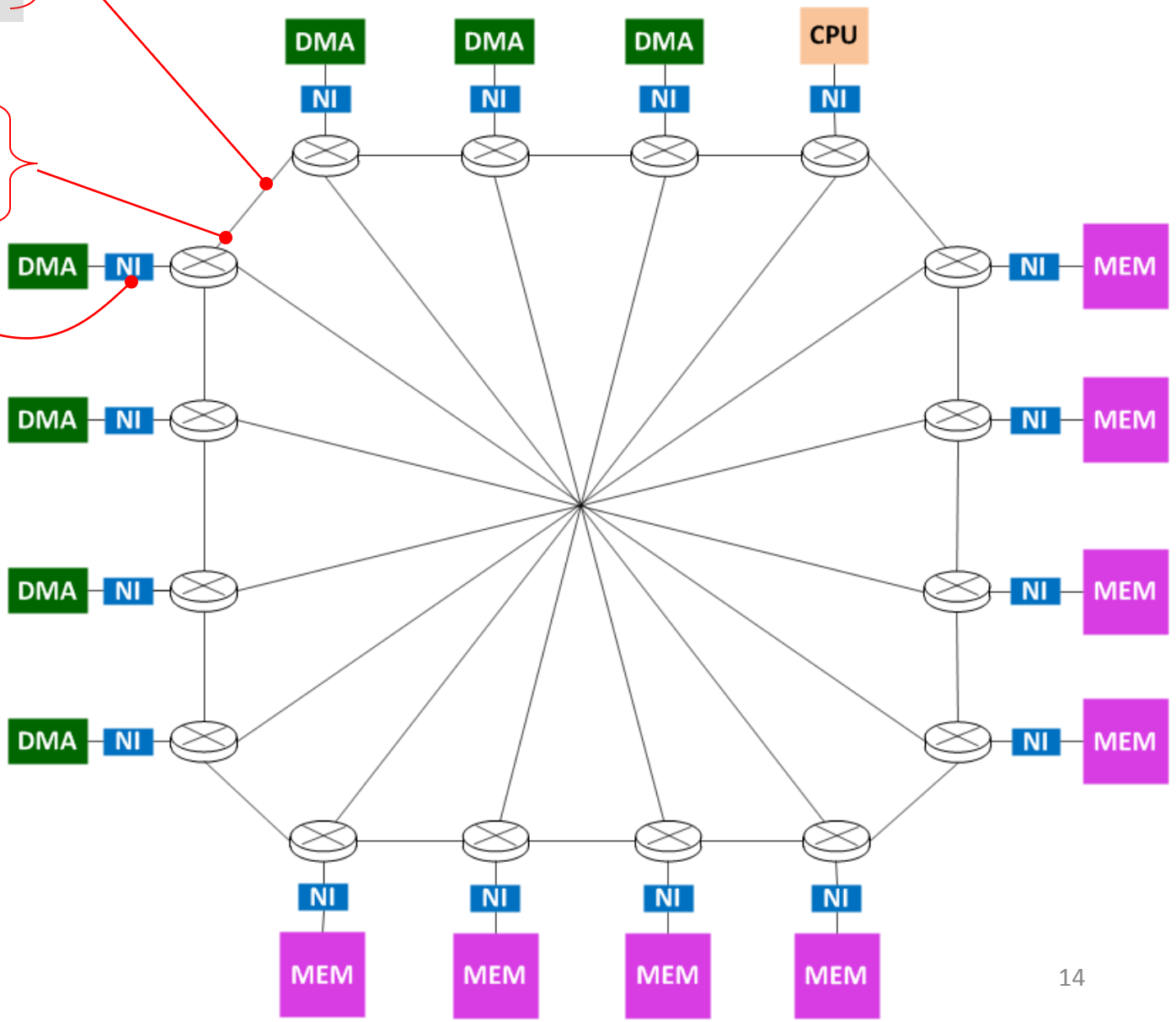

DMA
DMA
DMA
CPU
NI
NI
NI
NI
DMA
NI
NI
MEM
DMA
NI
NI
MEM
DMA
NI
NI
MEM
DMA
NI
NI
MEM
NI
NI
NI
NI
MEM
MEM
MEM
MEM

Benefits


32-B

VS.


Interlaving
at the INIs
+
8-B


DMA
DMA
DMA
CPU
NI
NI
NI
NI
DMA
NI
NI
MEM
DMA
NI
NI
MEM
DMA
NI
NI
MEM
DMA
NI
NI
MEM
NI
NI
NI
NI
MEM
MEM
MEM
MEM

Experimental system parameters

- Interleaving step
  - pace with which an initiator changes destination (0-64)
- Different amount of nodes and setups
  - #CPUs / #MEMs / #DMA Engines / #DDR2 Controllers
  - 8, 16, 32 nodes
- Packet injection rate
  - #packets per cycle per CPU
  - does not apply in DMA; a DMA engine initiates a transaction upon completion of its previous one
- Different link-widths
  - 8, 16, 32 Bytes
- Packet size=128 Bytes, #VCs=2, FBA is disabled

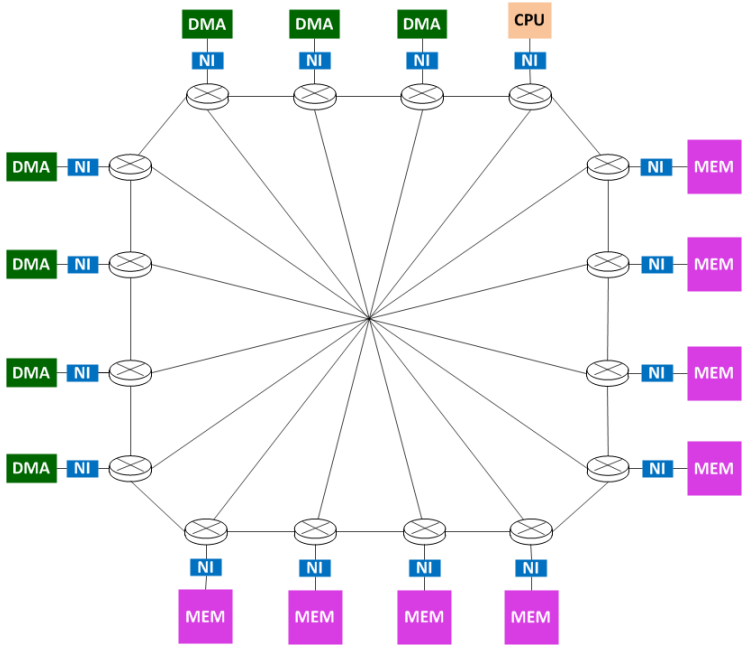
Experimental framework

- gem5 Simulator
  - detailed memory systems and interconnect models
- ..enhanced with Spidergon STNoC
  - time-annotated from RTL
- ..to measure the effect of address interleaving
  - end-to-end delivery times, i.e. NoC transfer delay
  - throughput
  - power consumption

Setup with CPUs

- 8 nodes: 4 CPUs and 1/2/4 channels
- Different packet injection rates
- Interleaving step: 1, 2, 4,…64
- Link-width: 16 Bytes

- Aim of study:
  - best interleaving step
  - saturation point
  - #channels



Saturation point

- Reason of saturation is congestion
  - buffers are filled-up, problem propagates to entire network
  - system unable to break-down the packets within reasonable time
- Range of packet injection rate – per clock cycle for each CPU – beyond which NoC saturates
  - …when NoC delay changes dramatically, e.g. from 60 to 500 cycles when increasing slightly the rate (+0.005 p/cc/CPU)
  - saturation begins when injection rate is 0.025-0.03 [1]

[1] *K. Papadimitriou, P. Petrakis, M. D. Grammatikakis, M. Coppola, "**Security enhancements for building saturation-free, low-power NoC-based MPSoCs**" , in Proc. CNS 2015, pp. 594-600*

Results from experiments with CPUs

# channels	*packet injection rate (w/o causing NoC saturation)*	*improvement*
w/o interleaving	*0.03*	-
2 channels	*0.05*	***1.66x***
4 channels	*0.07*	***2.33x***

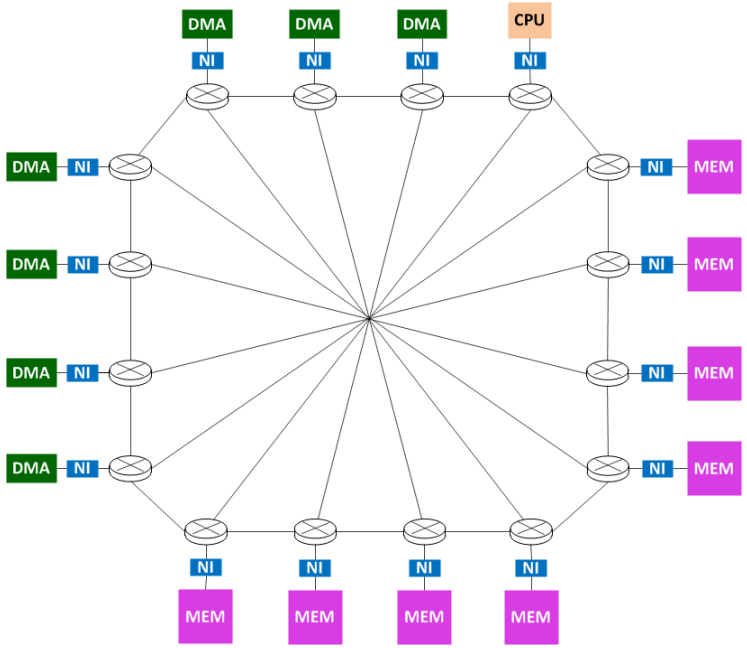
- Best interleaving step value is 1
  - change destination node after every transaction
- Address interleaving
  - reduces transfer delay on NoC
  - remedies NoC saturation
  - allows for higher injection rates
- Improvement increases with the #channels

we observe: interleaving can result in balancing effectively network load, and in less busy queues

Setup with DMAs

- 16 nodes: 7 DMA engines and 1/2/4/8 channels
- 32 nodes: 23 DMA engines and 1/2/4/8 channels
- Link-width: 8,16,32 Bytes
- Interleaving step: 1
  - most effective value

- Aim of study:
  - best link-width (LW)
  - aggregate throughput



Results from experiments with DMAs

	Aggregate throughput (MB/sec)		
# channels	**LW = 8 Bytes**	**LW = 16 Bytes**	**LW = 32 Bytes**
1 (w/o interleaving)	**10,710**	**15,119**	**15,960**
2	16,181	18,306	18,882
4	17,676	18,850	19,264
8	**18,170**	**19,056**	**19,280**
max improvement	**69,64%**	26,04%	20.8%

DMA transfer = 128 Bytes

- Address interleaving
  - allows for configuring NoC with narrow link-width
  - increases aggregate throughput
- Improvement increases with the *#channels*
  - we noticed this before (in experiments with CPUs)

Results from experiments with DMAs

	Router power (watt)		
# channels	**LW = 8 Bytes**	**LW = 16 Bytes**	**LW = 32 Bytes**
1 (w/o interleaving)	**1,194**	**1,721**	**2,759**
2	1,274	1,769	2,823
4	1,307	1,786	2,845
8	**1,321**	**1,798**	**2,856**
max power overhead added	**10,63%**	4,47%	3,51%

- Router's clock power doubles when link-width doubles
- Narrow link-width of NoC allows for power savings at the routers

Summary - Effect of interleaving

- Address interleaving
  - allows for configuring NoC with 8-Bytes link-width
  - reduces transfer delay on NoC
  - remedies NoC saturation
  - allows for higher injection rates
  - increases aggregate throughput
- Best interleaving step value is 1
  - change destination node per transaction
- Improvement increases with the #channels

we observed: interleaving can result in balancing effectively network load and in less busy queues

TEI of Crete

- more info
  - kpapadim@cs.teicrete.gr
  - mdgramma@cs.teicrete.gr
- work done in the context of DREAMS project

  http://www.dreams-project.eu/
- extensions added in gem5

  http://www.m5sim.org/Publications