

# Three tips to maximize your SoC performance

**ARM**

Tom Conway  
Director of Systems Marketing

Slides presented 24 Jan 17

<https://developer.arm.com/products/system-design/system-guidance/webinar-three-tips-for-maximising-your-soc-performance>

Jan 2017

©ARM 2017

# In this talk

- Introduce more about our work in system guidance
- Summarize some of the performance analysis methodologies we use inside ARM
- Discuss our “Three top tips” for maximizing SoC performance
- Present some of our example results

# System guidance

# Why provide system guidance?



Faster time  
to market



Reduced risk

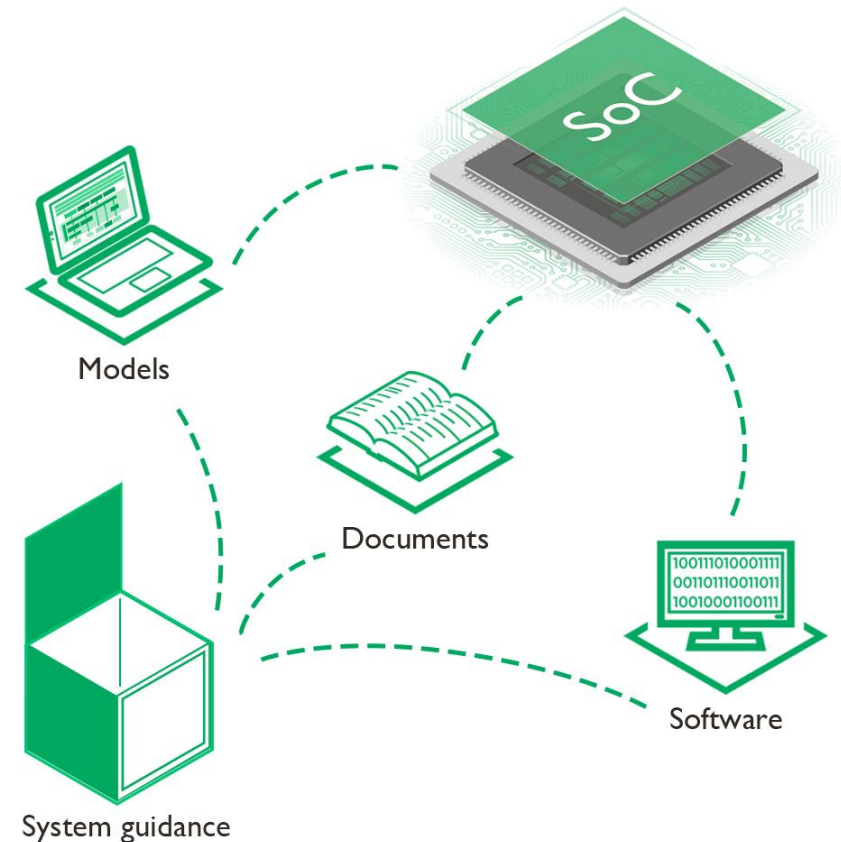


The ARM solution

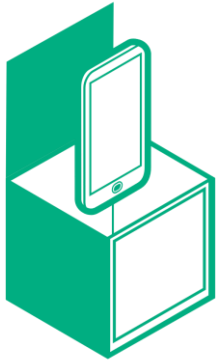
- Reduce partner time-to-market
- Reduced risk
- Show what you can achieve with ARM IP

# What is system guidance?

- System guidance is a collection of free resources to provide a view of typical compute subsystems that can be created using ARM IP
- System guidance comprises of:
  - Documents
    - System design presentations
    - Technical overview
    - System analysis report
    - Implementation guidelines
    - FVP programmers guide
  - Fixed Virtual Platform
  - Software
    - Build scripts and patches for open source

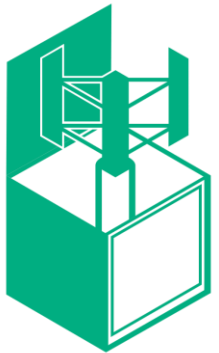


# System guidance for different markets



- **Mobile computing**

- CoreLink SGM-772: Cortex-A72 + Cortex-A53 + Mali-T880
- CoreLink SGM-773: Cortex-A73 + Cortex-A53 + Mali-G71
- CoreLink SGM-573: Cortex-A73 + Cortex-A53 + Mali-T820



- **Infrastructure (servers and networking)**

- CoreLink SGI-572: Cortex-A72 or Cortex-A53 + CoreLink CCN-512



- **IoT endpoints**

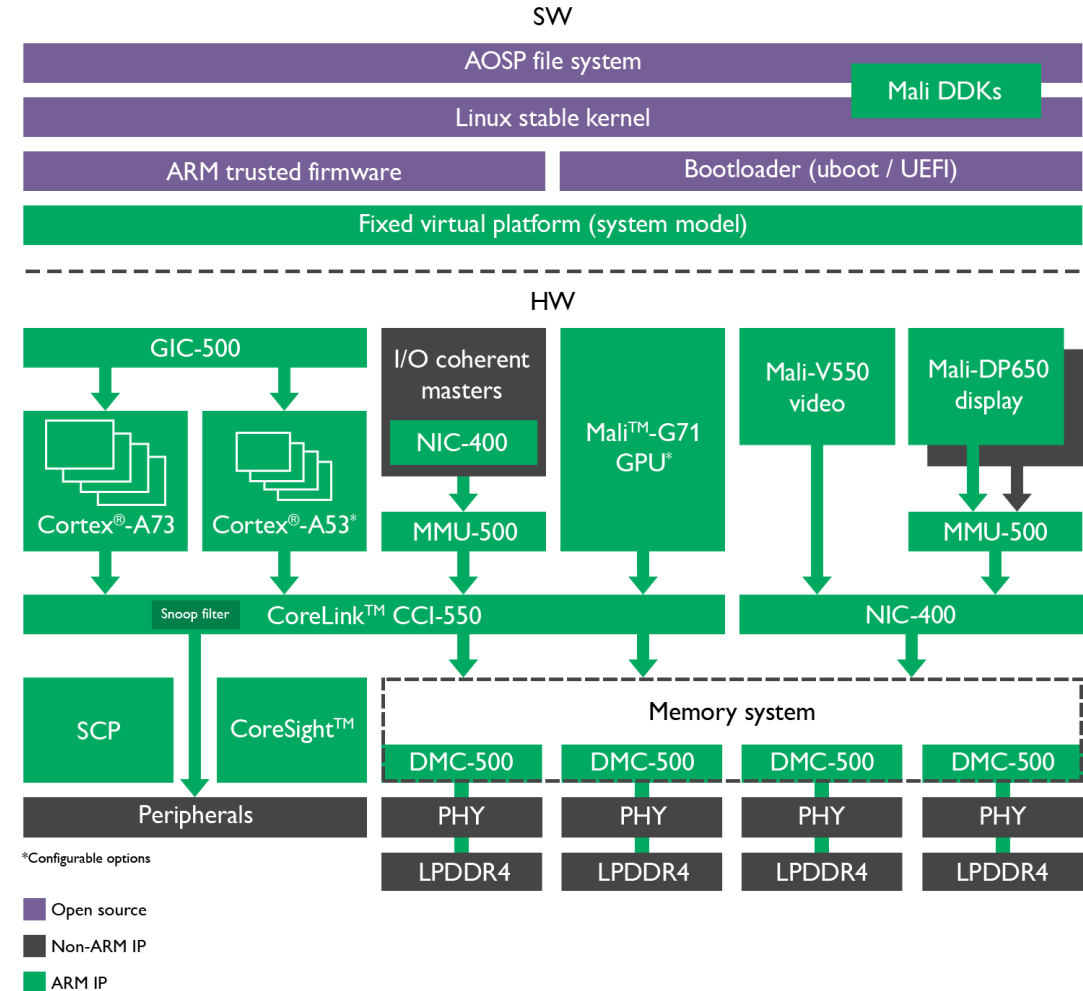
- Subsystem products available to license CoreLink SSE-200
- System Guidance coming soon for Cortex-M33

- **More coming in 2017**

# System Guidance for Mobile

# CoreLink SGM-773

- Target premium mobile
  - Released in Q2 2016, describes 2017 solutions
- ARMv8-A hardware
  - big.LITTLE architecture
  - Octa-core Cortex-A73 & Cortex-A53
  - Fully coherent Mali-G71 (MPI2)
  - Secure payment and content protection
  - Ultra HD media integration (CODEC & Display)
  - Implementation guidelines for 16nm FinFet
- Software stack
  - ARM Trusted firmware enabling security
  - Linux Kernel
  - Example media integration into Android

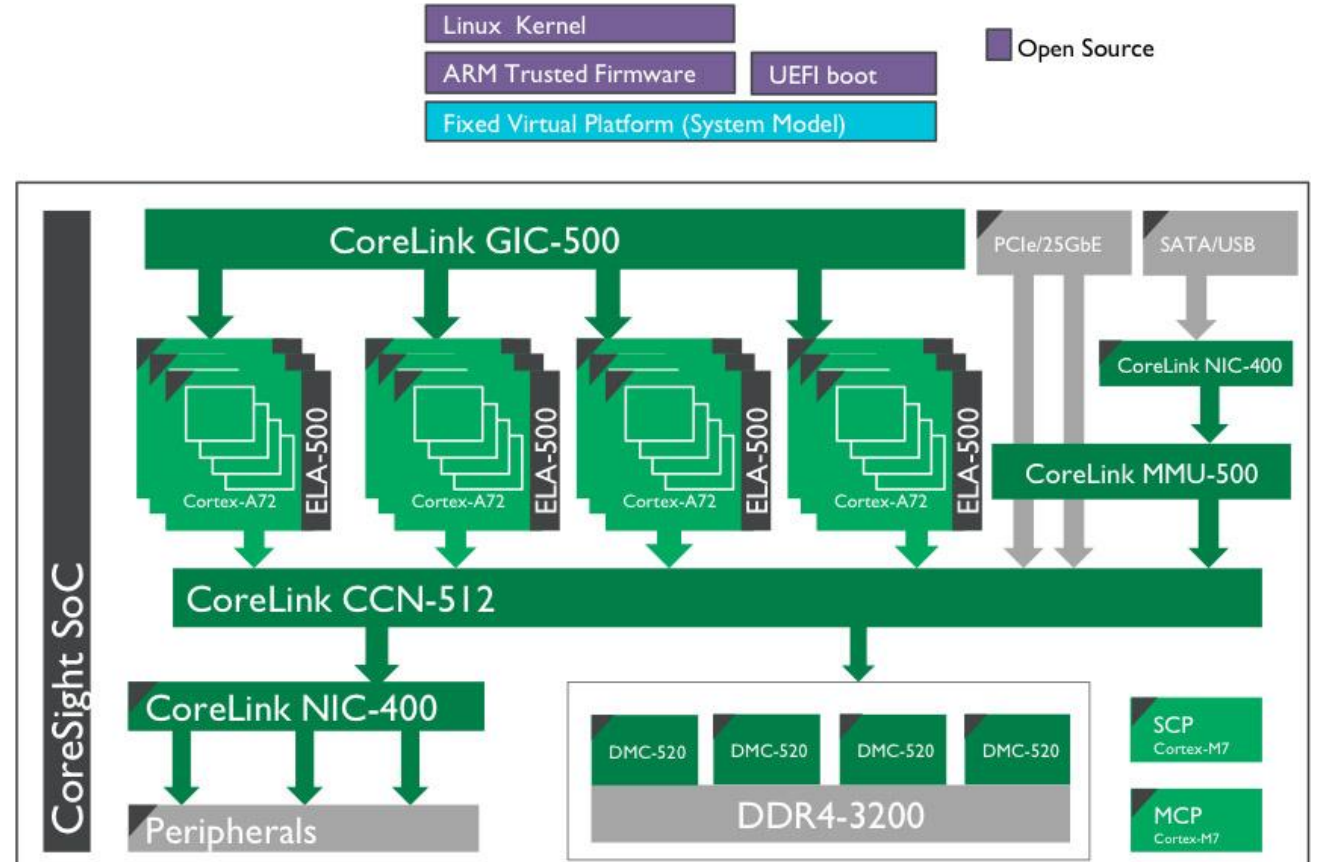




# System Guidance for Infrastructure

# CoreLink SGI-572

- Target low-to-mid range infrastructure
  - Released in 2016, describes 2017 solutions
- ARMv8-A hardware
  - Server Based System Architecture (SBSAv3)
  - 48x Cortex-A72 or 24x Cortex A53
  - Up to 32MB L3 cache
  - Up to 4x DDR4-3200 (DMC-520)
  - Up to 18 AXI expansion ports
  - I/O Coherent PCIe, Ethernet, accelerators
- Software stack
  - Standard platform interfaces with ARM Trusted Firmware
  - Linux kernel



# Poll question

# Poll question

- What pre-silicon analysis techniques do you rely on for your SoC performance verification?
- A. RTL simulation
- B. Performance modelling
- C. RTL emulation
- D. RTL simulation & emulation
- E. All of the above

One CPU generation's worth of performance with  
improved CPU configurations  
and  
optimized system architecture

# Performance analysis

# Deeper dive into performance analysis

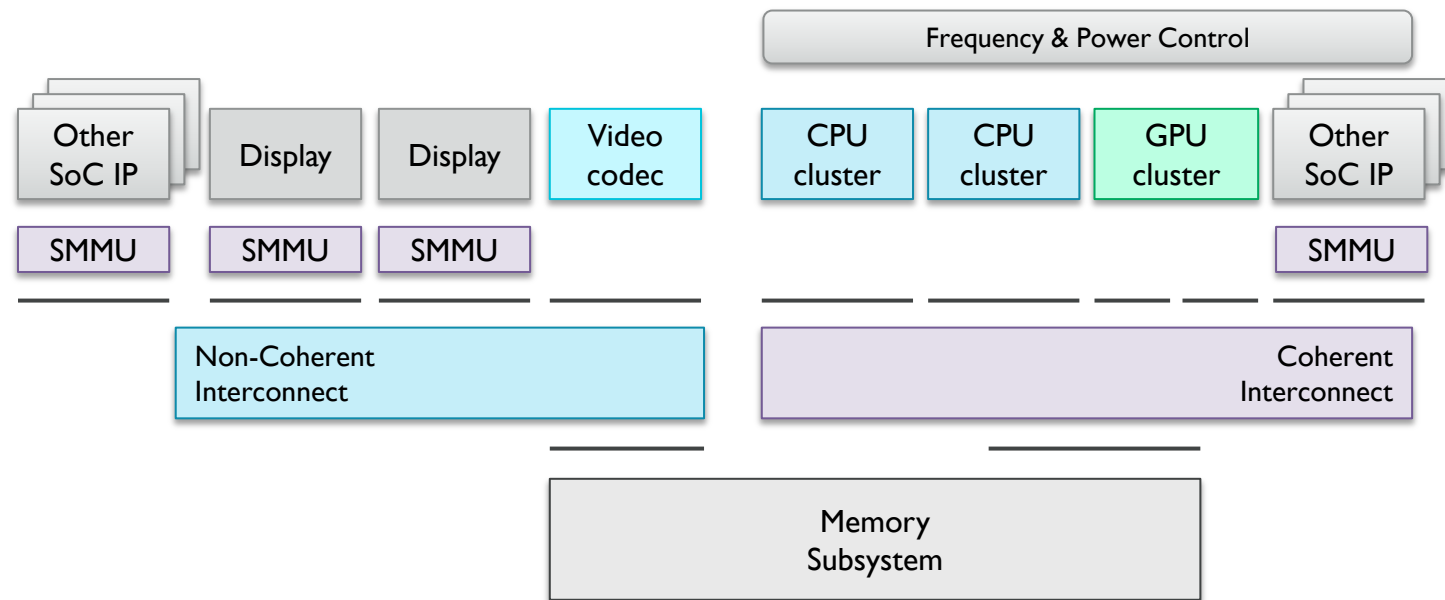
- Summarise some of the methodologies we use inside ARM
- Discuss our “Three top tips” for maximising SoC performance
- Present some of our example results

# Analysis platforms

- ARM conducts analysis at various levels of hardware abstraction
  - Static analysis
  - Cycle approximate modelling
  - RTL simulation
  - RTL emulation
  - FPGA
  - Silicon
- Software testing from bare metal up to full operating system
  - Internal micro-benchmarks
  - Industry benchmarks



# System performance analysis – overview



## Benchmarking (OS)

Linux CPU benchmarks  
Android CPU benchmarks

## System use-cases (bare-metal)

System use-cases (CPU+Media+System IP)

## IP system performance (bare-metal)

CPU memory performance (+ big.LITTLE)  
GPU graphics/compute performance (+ CPU/GPU)  
Video performance  
Display performance without/with SMMU

## Traffic analysis (VPE)

Bandwidth, Latency, DDR Efficiency, Coherency  
Use-cases, Quality of Service  
SMMU – IP traffic, frame buffer compression, rotation

# Deeper dive into performance analysis

- Summarise some of the methodologies we use inside ARM
- Discuss our “Three top tips” for maximising SoC performance
- Present some of our example results

# Three top tips for SoC performance

## 1 Minimize CPU path to memory

Every clock cycle counts for CPU latency

## 2 Maximize system bandwidth

Ensuring the precious system memory is maximized for performance

## 3 Manage traffic types

Give CPU priority, while still meeting real time contracts

# I. Minimise CPU path to memory

Topics to consider:

Four points to consider

- SoC interconnect topology .....
- CPU physical placement and floor planning .....
- Minimise clock crossings .....
- CPU issue capability .....

Ia

Ib

Ic

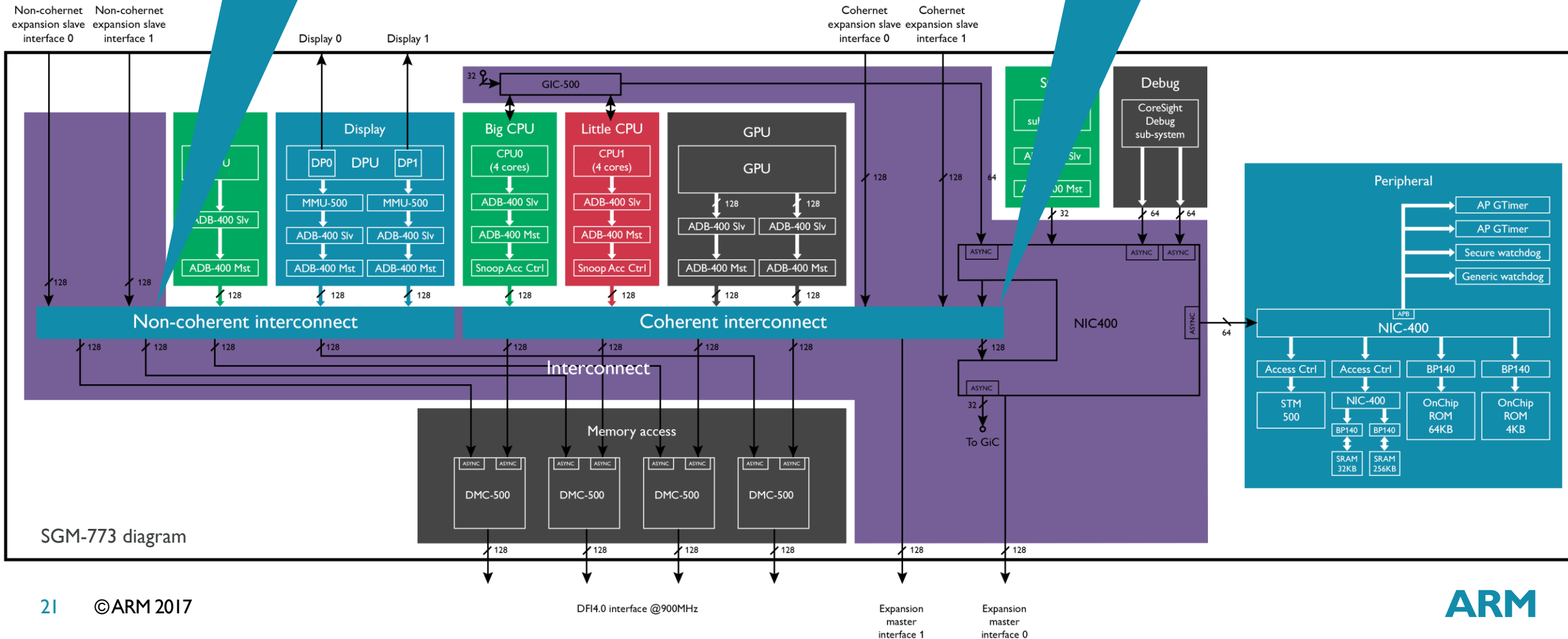
Id

# SoC interconnect topology

1a

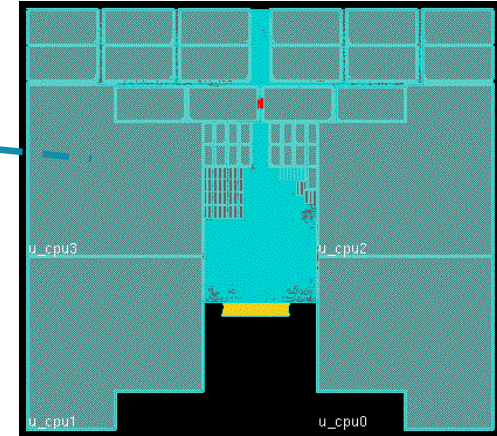
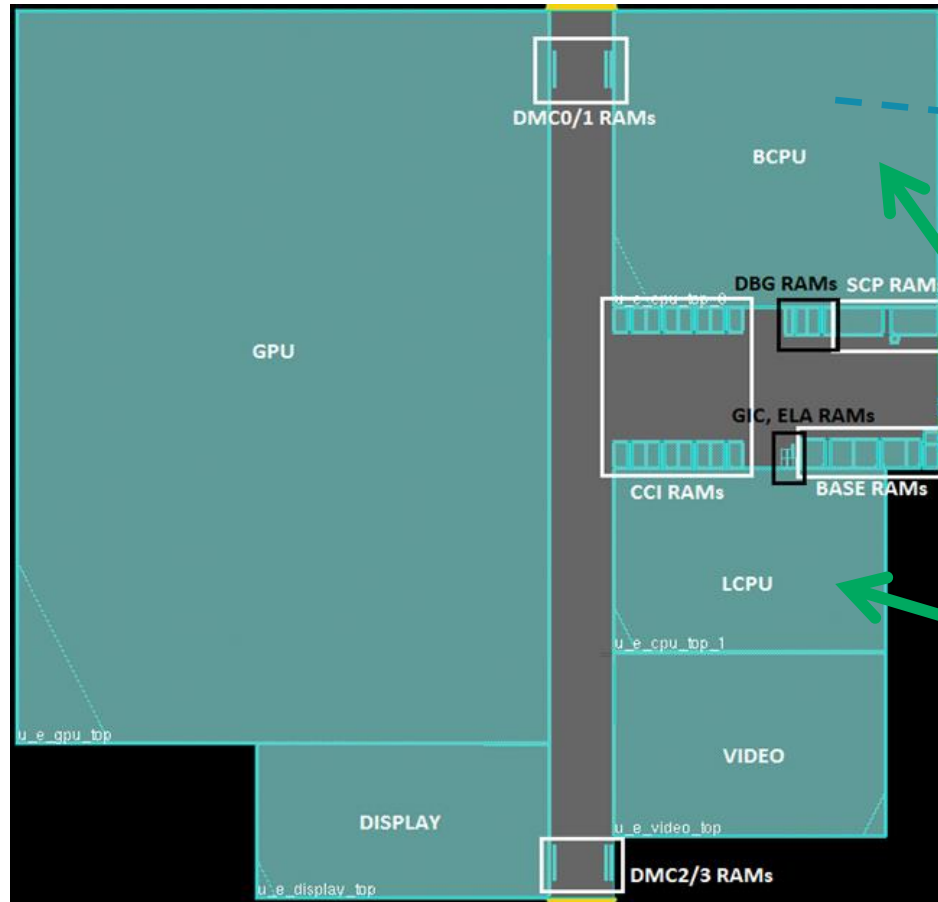
Non-coherent path to memory  
for media real-time traffic

Coherent path to memory for  
CPU/GPU



# CPU physical placement and floor planning

1b



\*\* Layout taken from SGM-773 implementation guidelines

CPU clusters placed closest  
to interconnect

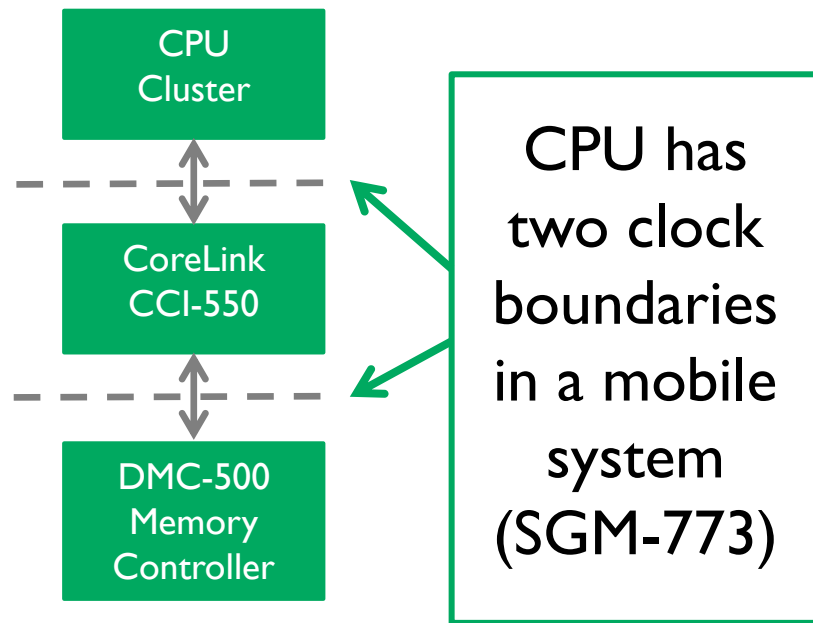
Thus minimizing register retiming stages

\*\* Floorplan is optimised for Package on Package

# Minimise clock crossing

lc

- Each processor cluster has separate DVFS domain
- Each clock domain crossing costs a minimum of 3-cycles if pipelined
- Clock domain crossing costs you twice – forward and return path



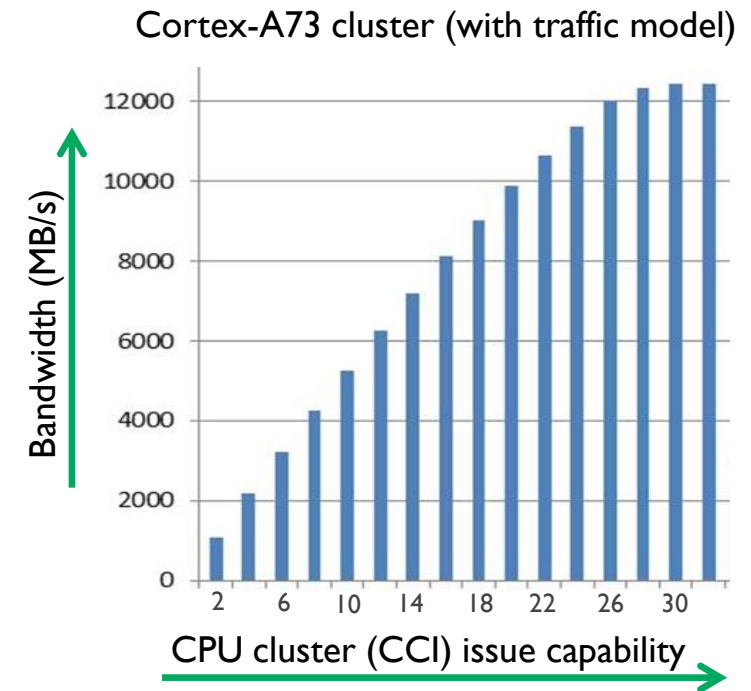
Components	SUD 0v6	UD 0v7	NOM 0v8	OD 0v9	SOD 1.0
Cortex®-A73 processor	-	1.31GHz	1.75GHz	2.1GHz	2.45GHz
Cortex-A53 processor	675MHz	1GHz	1.35GHz	1.6GHz	-
Mali-G71	425MHz	637.5MHz	850MHz	-	-
Mali-DP650	-	-	650MHz	-	-
Mali-V550	-	-	600MHz	-	-
CoreLink CCI-500 Cache Coherent Interconnect	-	-	800MHz	-	-
CoreLink DMC-500 Dynamic Memory Controller	-	-	800MHz	-	-

\*\* Table taken from SGM-773 implementation guidelines

# CPU issue capability

Id

- Each ARM CPU cluster has a specified issue capability
  - Determined by Fetch and Evict Queue depth (FEQ depth) of CPU cluster
  - CPU latency is not just about clock cycles
  - Greater CPU issue capability also improves effective latency
- SoC interconnect must be sized accordingly
  - Configure CCI interconnect to maximize... CPU issue capability
  - But do not oversize beyond system latency



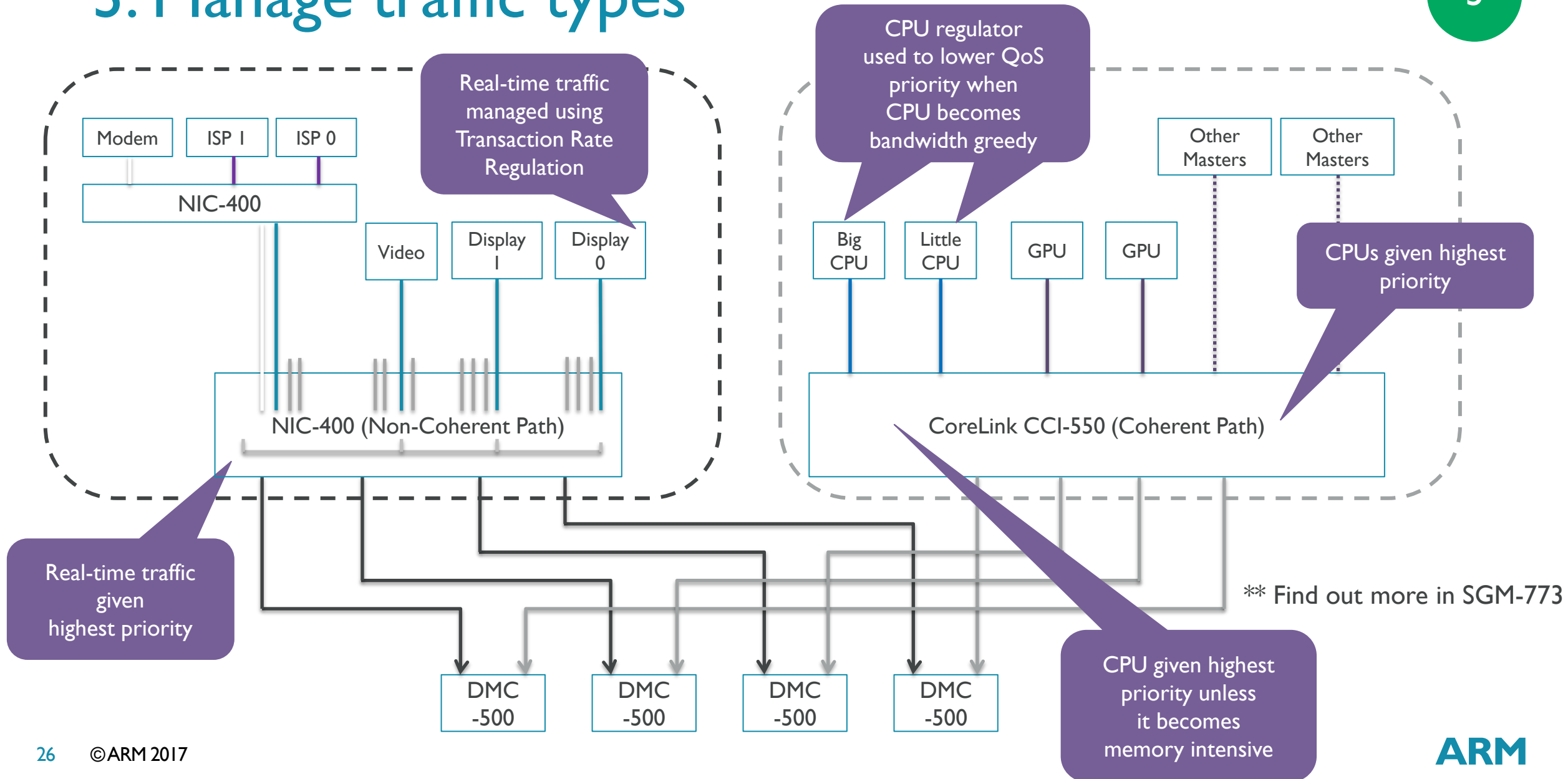


## 2. Maximize SoC bandwidth

- Fine-grained address interleaving across memory controllers
  - Improves performance due to improved load balancing of memory resources
  - Higher bandwidth & lower latency
- Performance & power trade-off
  - Trade-off is that there may be fewer in-row hits in the DMC with smaller stripe sizes, this may lead to increased power consumption
- Address hash can be used to avoid traffic hot spotting
  - This can include AFBC compressed traffic and media masters with x-y decoder schemes

\*\* Find out more in SGM-773

# 3. Manage traffic types



# Deeper dive into performance analysis

- Summarise some of the methodologies we use inside ARM
- Discuss our “Three top tips” for maximising SoC performance
- Present some of our example results

# Performance results from SGM-773

- When looking at CPU latency, ARM uses various techniques:
  - Static latency ..... spreadsheet
  - Pointer chase ..... simulation
  - LMBench measurements ..... emulation
- These can be measured at different stages as the system design progresses

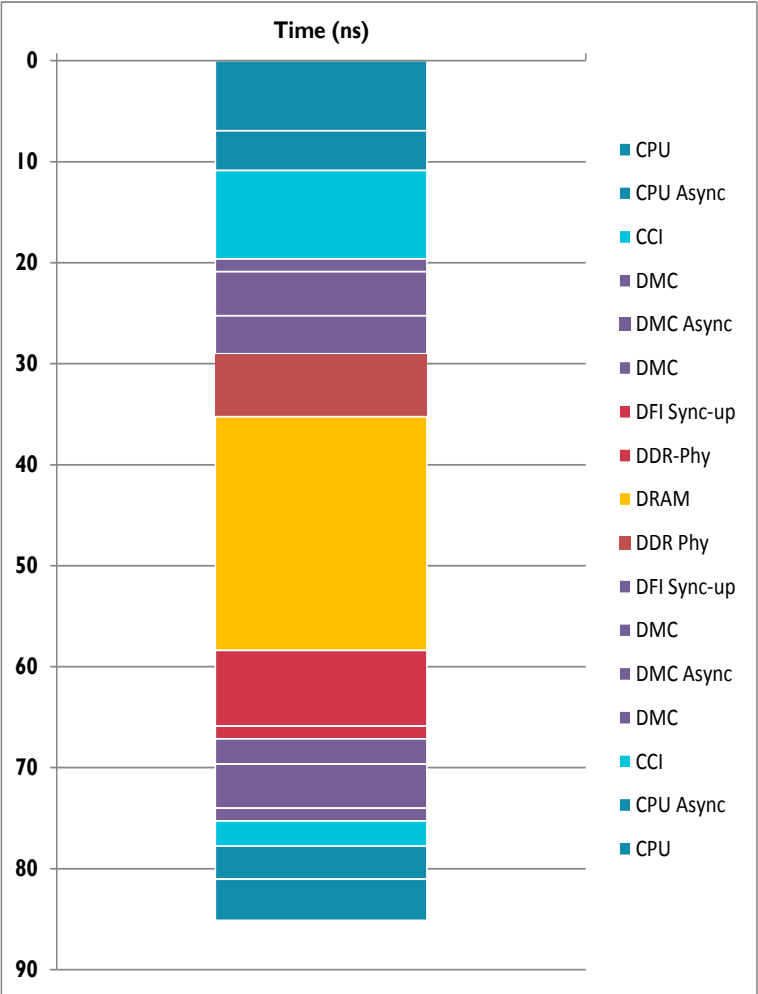
# Static latency results for SGM-773

Memory technology:  
4-channel LPDDR4-3200 @16-bit

“Open Row” DDR access

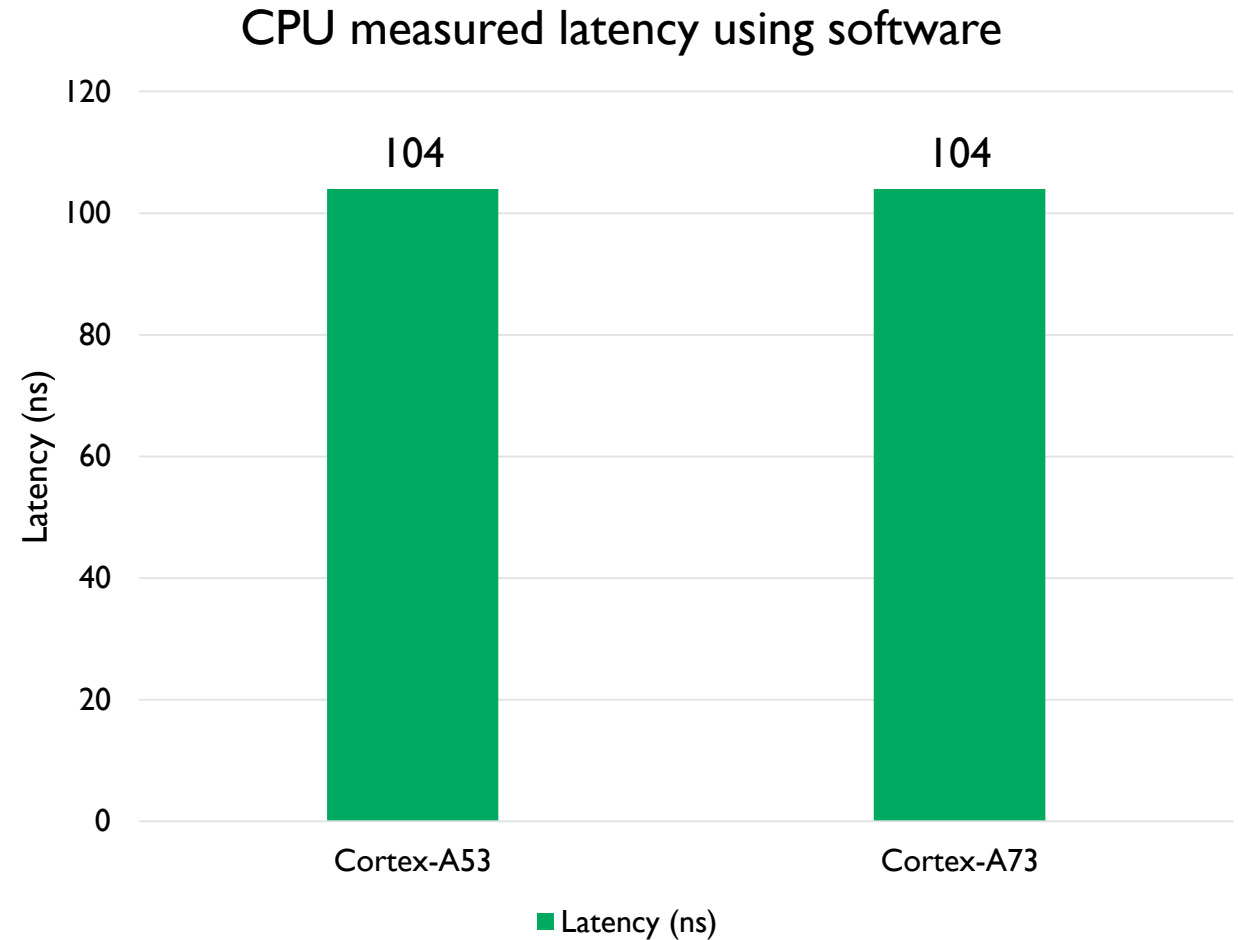
	IP	Cortex-A73	CPU Bus	CCI-550	DMC-500	Phy	DRAM	Total
	Freq (MHz)	2450	1225	800	800	1600	1600	
	Domain	No of cycles						Time (ns)
Outbound	CPU	15	1					6.9
	CPU Async		1	2.5				3.9
	CCI			7				8.8
	DMC			1				1.3
	DMC Async			1	2.5			4.4
	DMC				3			3.8
	DFI Sync-up				0	0		0.0
	DDR Phy					10		6.3
	DRAM						37	23.1
Inbound	DDR-Phy					12		7.5
	DFI Sync-up				1	0		1.3
	DMC				2			2.5
	DMC Async			2.5	1			4.4
	DMC			1				1.3
	CCI			2				2.5
	CPU Async		2.5	1				3.3
	CPU	8	1					4.1
	Subtotal (cycles)	23	5.5	18	9.5	22	37	

Spreadsheet static latency total: 85.1ns



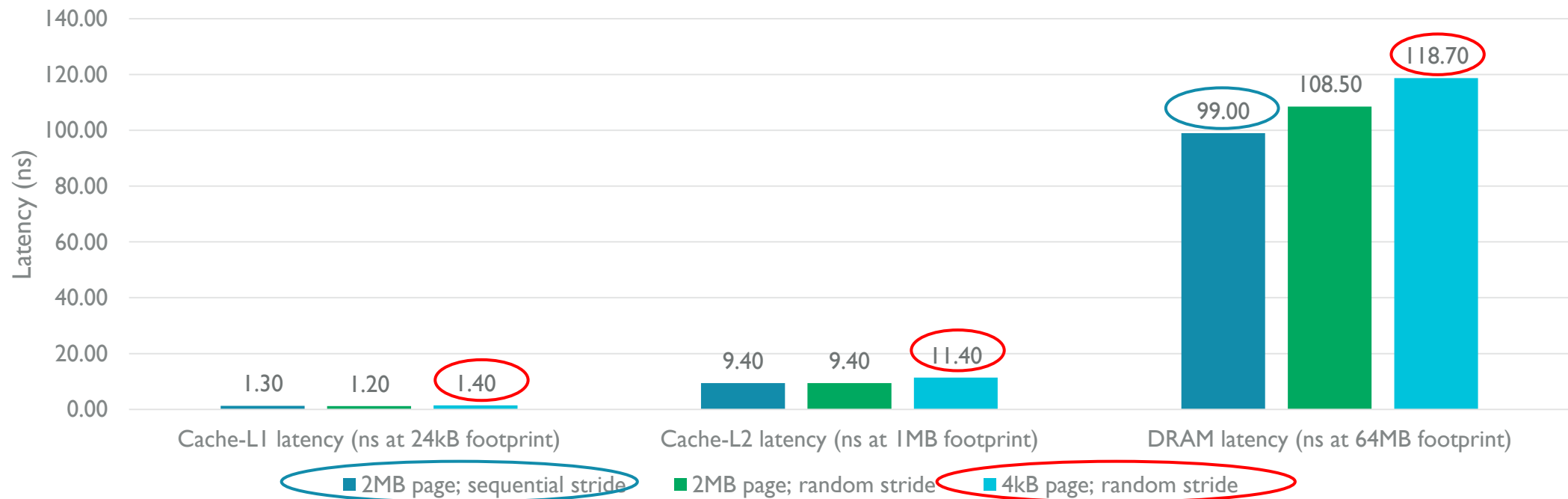
# Pointer chase results for SGM-773

- Pointer chase with bare-metal software
- CPU timer used to measure latency
- CPU configuration
  - MMU disabled
  - L1/L2 disabled
- Closed row DDR access



# LMBench results for SGM-773

- LMBench latency (bare-metal) for Cortex-A73
- Page size of 4KByte & 2MByte with either random address or sequential address

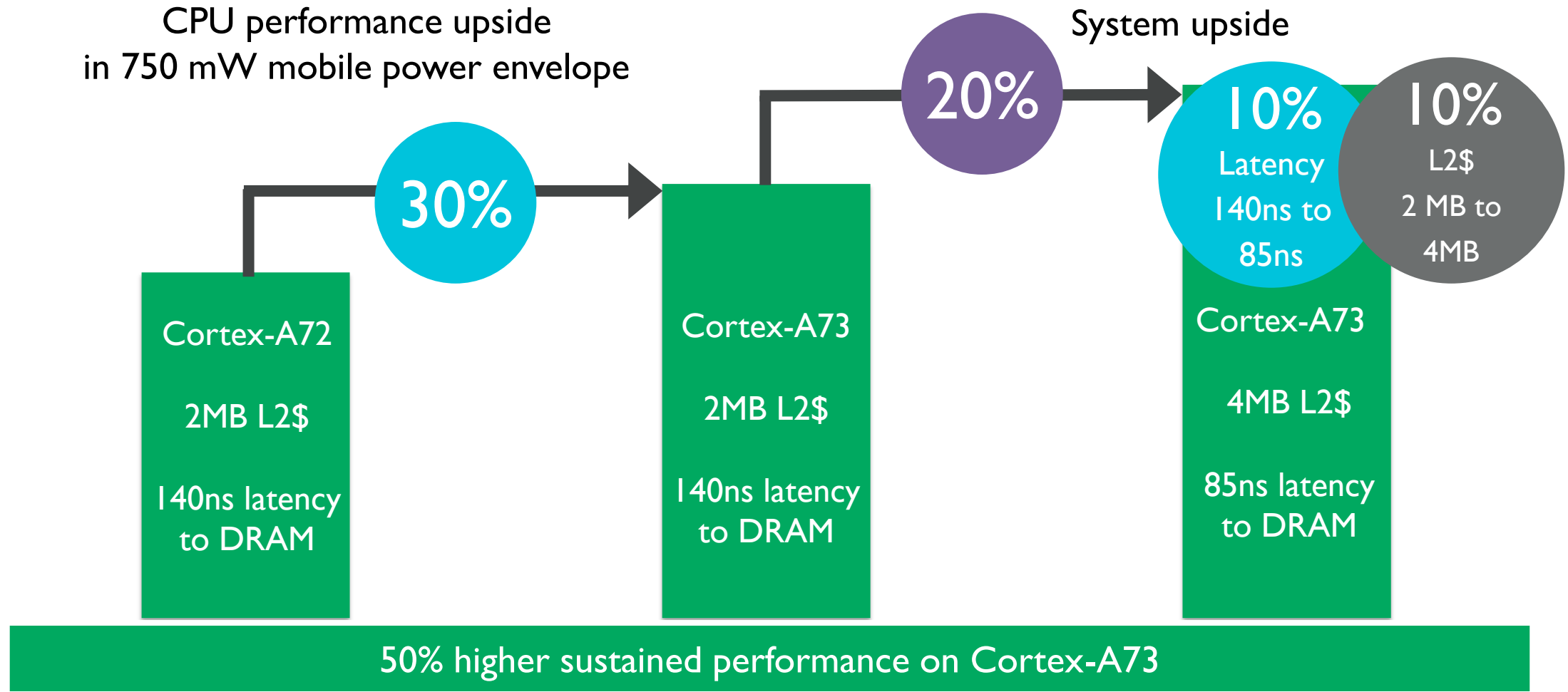


- Toolchain: GCC Linaro 4.9 2014.09
- Compiler Flags: `-g -pipe -mcpu=cortex-a57+crypto+crc -march=armv8-a+crypto+crc -O3 -fomit-frame-pointer`

One CPU generation's worth of performance with  
improved CPU configurations  
and  
optimized system architecture



# Maximizing sustained performance with Cortex-A73



# Summary

# Three top tips for SoC performance

1.

## Minimize CPU path to memory

Every clock cycle counts for CPU latency

2.

## Maximize system bandwidth

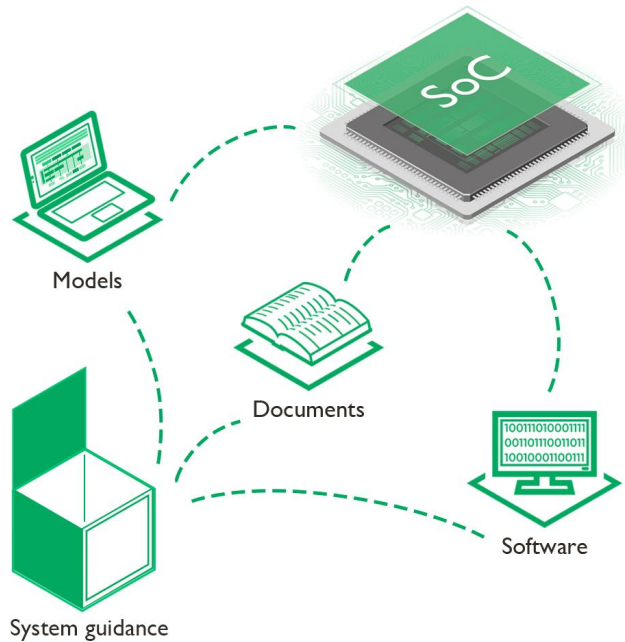
Ensuring the precious system memory is maximized for performance

3.

## Manage traffic types

Give CPU priority, while still meeting real time contracts

# System guidance – faster path SoC to market



Deliver SoC months  
earlier

Leverage ARM  
learnings

- System guidance provides free additional information to help partners design SoCs
- Comprehensive set of data helps you develop your SoC faster with less risk
  - Quickly assess performance targets to lock down your design quicker
  - Confidence that ARM IP works well together
  - HW and SW guidelines, open source patches
- Leverage ARM reference design key learnings
  - SoC integration challenges and performance analysis
  - Focus your efforts on differentiation

Questions? Want to know more?  
Please contact [tom.conway@arm.com](mailto:tom.conway@arm.com)



Or search 'system guidance' on  
[developer.arm.com](http://developer.arm.com)

# ARM

The trademarks featured in this presentation are registered and/or unregistered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

Copyright © 2017 ARM Limited

©ARM 2017