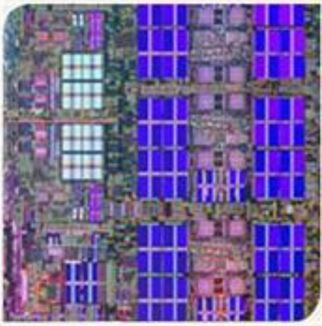
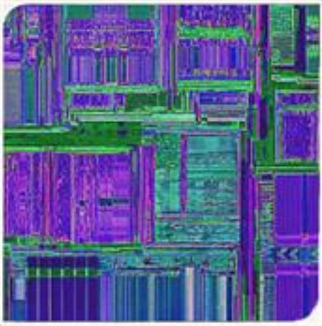




CS4803DGC Design and Programming of Game Console

Spring 2011

Prof. Hyesoon Kim

Georgia Tech College of Computing

Review: Exception Return

- After the exception handler, the hardware just starts from the user mode.
- Software must
  - Restore the modified registers
  - CPSR must be restored from the appropriate SPSR
  - PC must be changed back to the relevant instruction address in the user instruction stream
    - These two cannot happen independently

Return Address

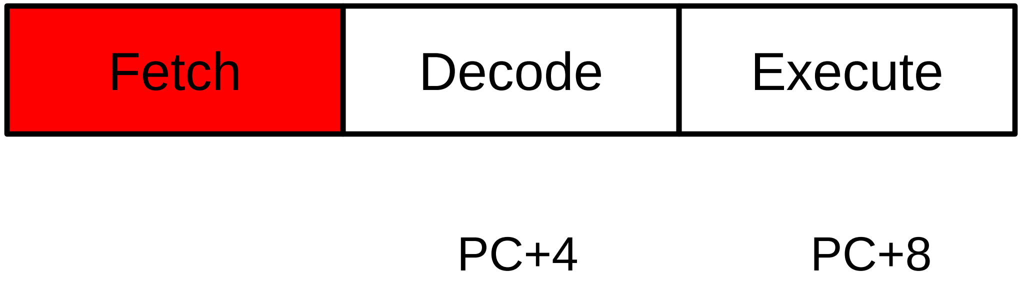
- IRQ and FIQ must return one instruction early in order to execute the instruction that raised an exception
- Prefetch abort must return one instruction early to execute the instruction that had caused a memory fault when first requested
- Data abort must return the instruction that caused exception.

Use of R15

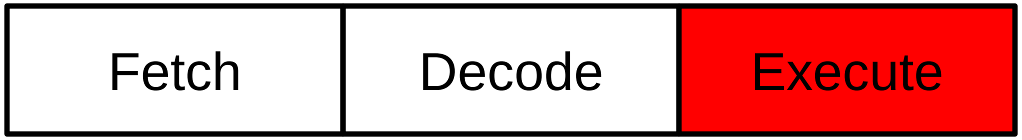
- R15: PC
  - PC may be used as a source operand
  - Register-based shift cannot use R15 as source operands.
- Running-ahead PC's behavior
  - PC is always running ahead
  - PC is always pointing +8 of the current instruction
    - Imagine 3-stage pipeline machine . PC is pointing what to fetch when an instruction is in the WB stage in the 3-stage pipeline machine
- When R15 is a source, the current PC + 8 is supplied to the source operand.
- When R15 is a destination
  - S: 1: SPSR→ CPSR, affecting interrupt, resource PC and CPSR automatically,

Exception generation time

- Pre-fetch abort : instruction fetch



- Data abort : memory execution



Controlling Interrupts

```
void event_EnableIRQ (void)
{
__asm {
            MRS r1, CPSR
            BIC r1, r1, #0x80
            MSR CPSR_c, r1
    }
}
// Enable Bit 7 (set register 0)
```

```
void event_DisableIRQ (void)
{
__asm {
            MRS r1, CPSR
            ORR r1, r1, #0x80
            MSR CPSR_c, r1
            }
}

   // Disable bit 7 (set 1)
```

31 28	27 8	7 6	5	4 0
N Z C V	**unused**	**IF**	**T**	**mode**

Bit 7: interrupt
Bit 6: Fast interrupt

Typical example of interrupt handler

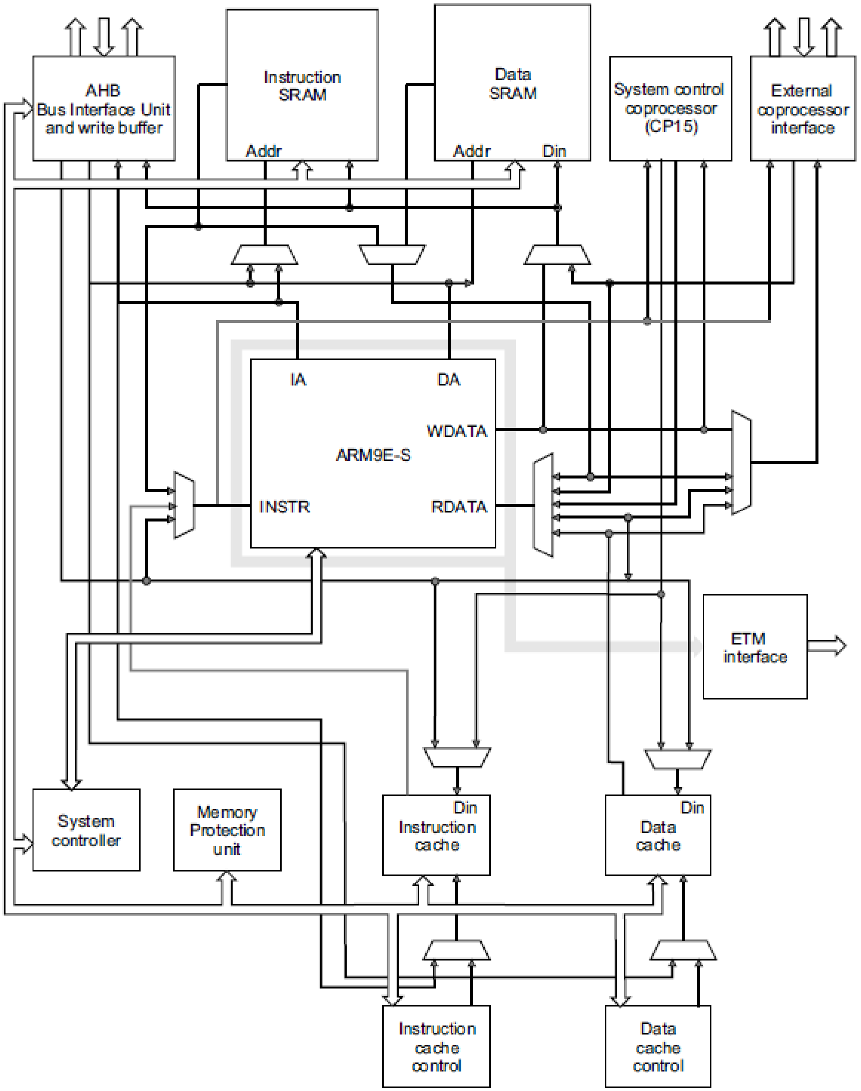
- SUB lr, lr, #4
- STMFD sp!{reglist, lr}

; ….

LDMFD sp!, {reglist,pc}^

ARM946E-S:(ARM 9 in Nintendo DS)

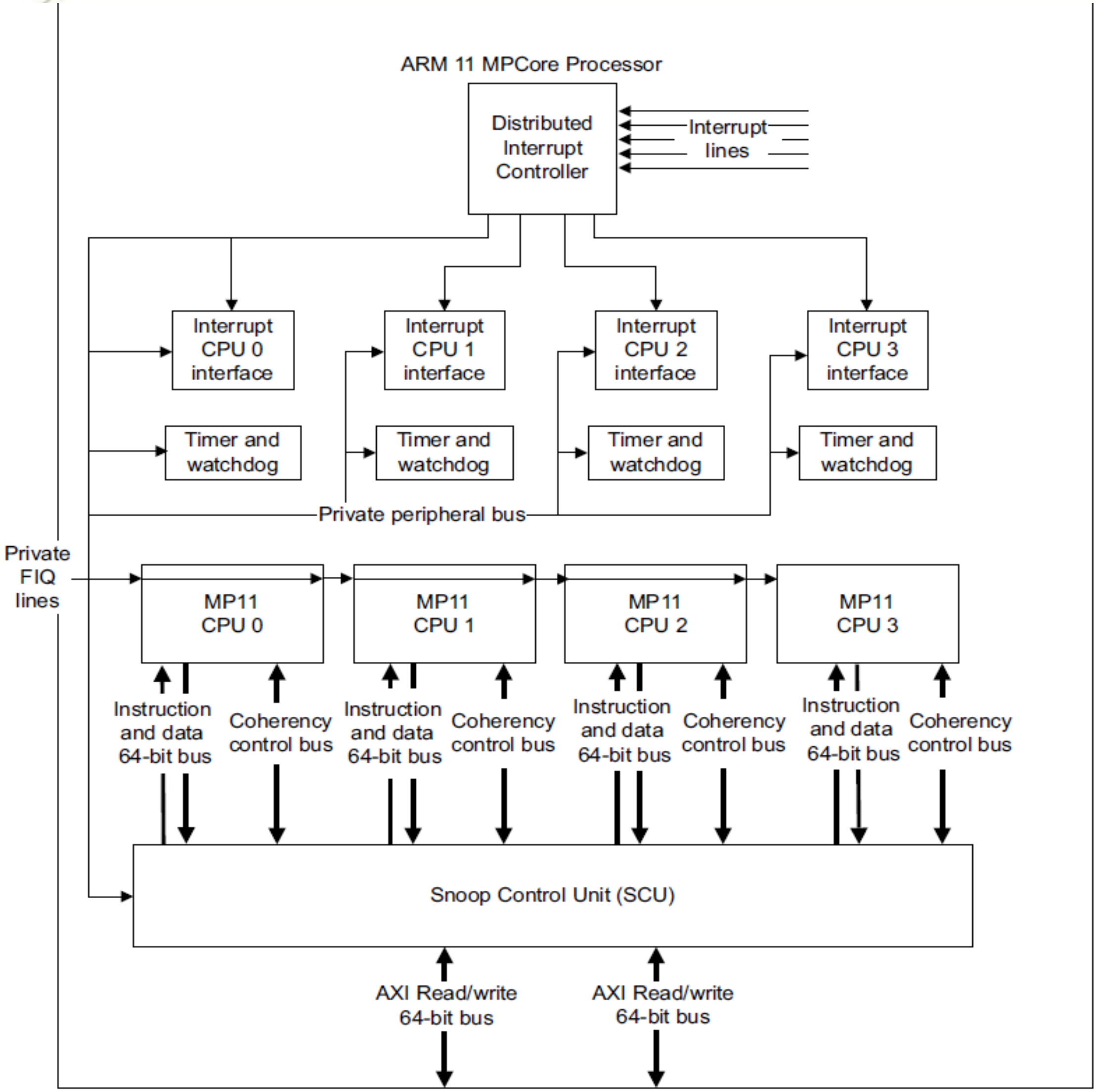
- Soc for embedded system.
- Single chip DSP
- Embedded applications running an RTOS
- Mass storage - HDD & DVD
- Speech coders
- Automotive control
  - Cruise control, ABS, etc.
- Hands-free interfaces
- Modems and soft-modems
- Audio decoding
- Dolby AC3 digital
- MPEG MP3 audio
- Speech recognition and synthesis.



ARM946E-S:(ARM 9 in Nintendo DS): Instructions

- Data processing instructions
- Load and store instructions
- Branch instructions
- Coprocessor instructions
  - Coprocessor data processing
  - Coprocessor register transfer
  - Coprocessor data transfer

ARM 11 MP Core Processor

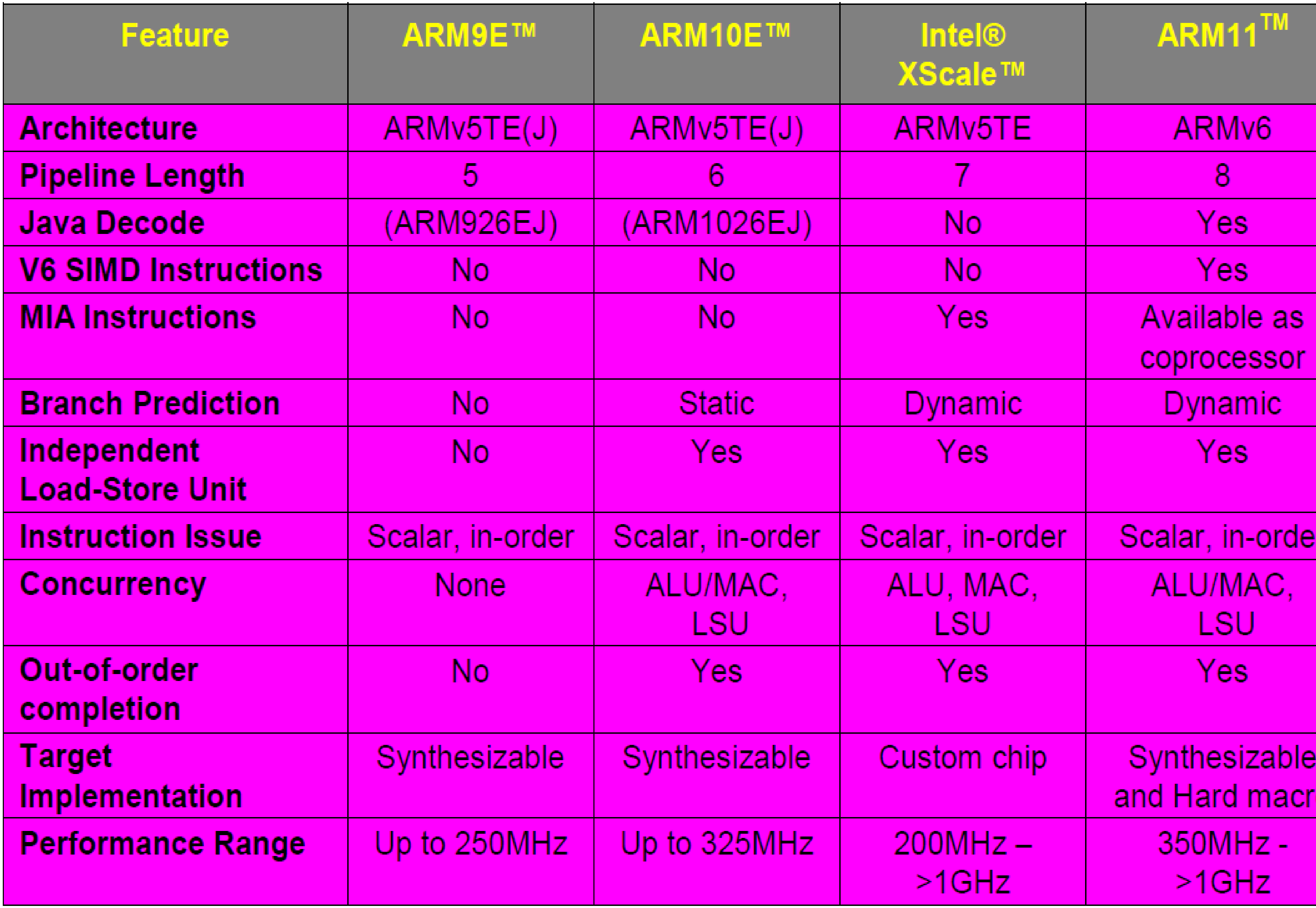

ARM 11 MPCore Processor
Distributed Interrupt Controller
Interrupt lines
Interrupt CPU 0 interface
Interrupt CPU 1 interface
Interrupt CPU 2 interface
Interrupt CPU 3 interface
Timer and watchdog
Timer and watchdog
Timer and watchdog
Timer and watchdog
Private peripheral bus
Private FIQ lines
MP11 CPU 0
MP11 CPU 1
MP11 CPU 2
MP11 CPU 3
Instruction and data 64-bit bus
Coherency control bus
Instruction and data 64-bit bus
Coherency control bus
Instruction and data 64-bit bus
Coherency control bus
Instruction and data 64-bit bus
Coherency control bus
Snoop Control Unit (SCU)
AXI Read/write 64-bit bus
AXI Read/write 64-bit bus

http://www.arm.com/files/pdf/ARMCortexA-9Processors.pdf

New Features in ARM 11

- Improve Memory Accesses
  - – Non-blocking (hit-under-miss) operations
- LD/ST and ALU are decoupled.
- Out-of-order completion:
  - – Instructions that have no dependency on the outcome of the previous instruction can complete. !!! → Good or Bad?

The ARM11 Microarchitecture, David Cormie, ARM Ltd



Feature	ARM9E™	ARM10E™	Intel® XScale™	ARM11™
Architecture	ARMv5TE(J)	ARMv5TE(J)	ARMv5TE	ARMv6
Pipeline Length	5	6	7	8
Java Decode	(ARM926EJ)	(ARM1026EJ)	No	Yes
V6 SIMD Instructions	No	No	No	Yes
MIA Instructions	No	No	Yes	Available as coprocessor
Branch Prediction	No	Static	Dynamic	Dynamic
Independent Load-Store Unit	No	Yes	Yes	Yes
Instruction Issue	Scalar, in-order	Scalar, in-order	Scalar, in-order	Scalar, in-orde
Concurrency	None	ALU/MAC, LSU	ALU, MAC, LSU	ALU/MAC, LSU
Out-of-order completion	No	Yes	Yes	Yes
Target Implementation	Synthesizable	Synthesizable	Custom chip	Synthesizable and Hard macr
Performance Range	Up to 250MHz	Up to 325MHz	200MHz – >1GHz	350MHz - >1GHz

Figure 5. ARM Architecture Feature Comparisons

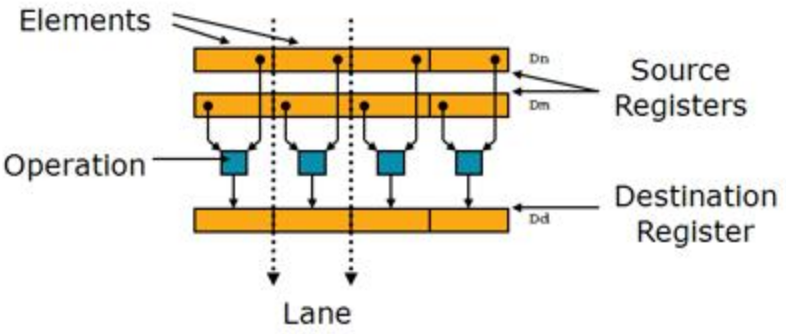
The ARM11 Microarchitecture, David Cormie, ARM Ltd.

Thumb-2 ISA

- Thumb-2 is a superset of the Thumb instruction set.
- Thumb-2 introduces 32-bit instructions that are intermixed with the 16-bit instructions. The Thumb-2 instruction set covers almost all the functionality of the ARM instruction set.
- Thumb-2 is backwards compatible with the ARMv6 Thumb instruction set.

SIMD in ARM

- Neon: ARM’s SIMD engine
- 128bit SIMD
- NEON instructions perform "Packed SIMD" processing:
- Registers are considered as **vectors** of **elements** of the same **data type**
- Data types can be: signed/unsigned 8-bit, 16-bit, 32-bit, 64-bit, single precision floating point
- Instructions perform the same **operation** in all **lanes**



http://www.arm.com/products/processors/technologies/neon.php

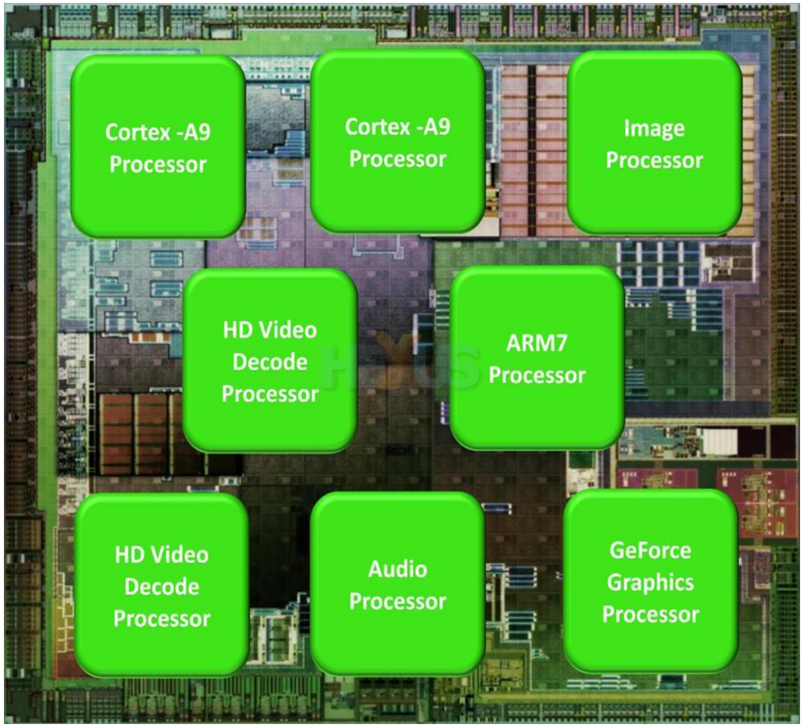
Usage model of NEON

- Watch any video in **any** format
- Edit and enhance captured videos - video stabilization
- Anti-aliased rendering and compositing
- Game processing
- Process multi-megapixel photos quickly
- Voice recognition
- Powerful multichannel hi-fi audio processing

http://www.arm.com/products/processors/technologies/neon.php

- Coretex A-9

Tegra 2 SoC



- Dual core: ARM Cortex-A9 processors
- Ultra Low Power GeForce GPU
- ARM 7 core
- Multimedia support
  - Audio, vidoe decode/encode

Slides from Kim and Kumar's presentation

Platforms using Tegra Boards

Notion Ink Adam

Motorola Xoom

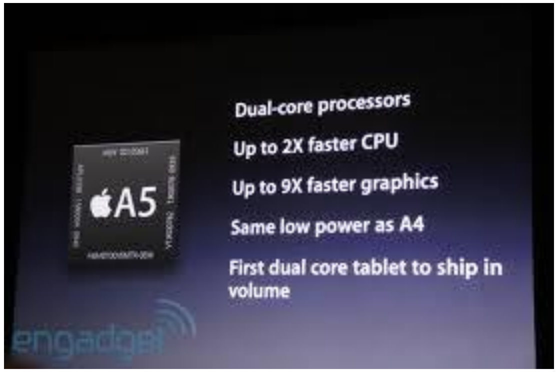
Viewsonic g Tablet

Motorola Atrix 4g

Slides from Kim and Kumar’s presentation

Ipad/Ipad2

- **dual Cortex A9, PowerVR SGX 543MP2**



Dual-core processors

Up to 2X faster CPU

Up to 9X faster graphics

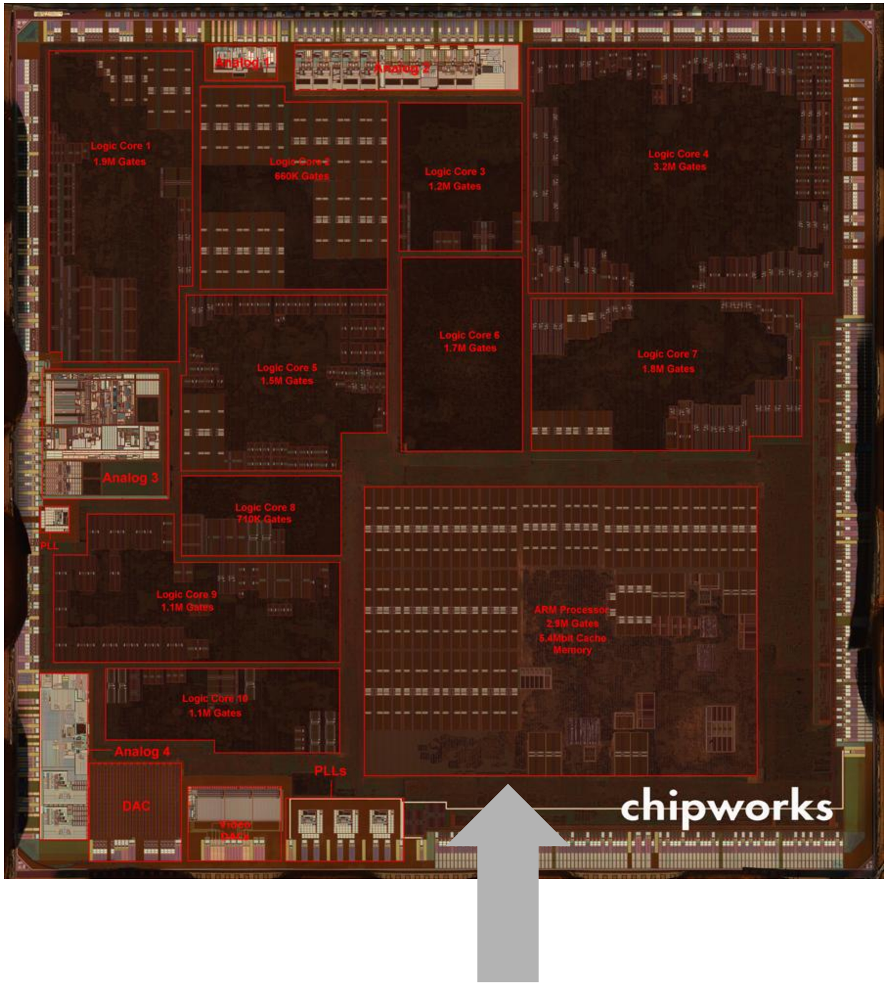
Same low power as A4

First dual core tablet to ship in volume

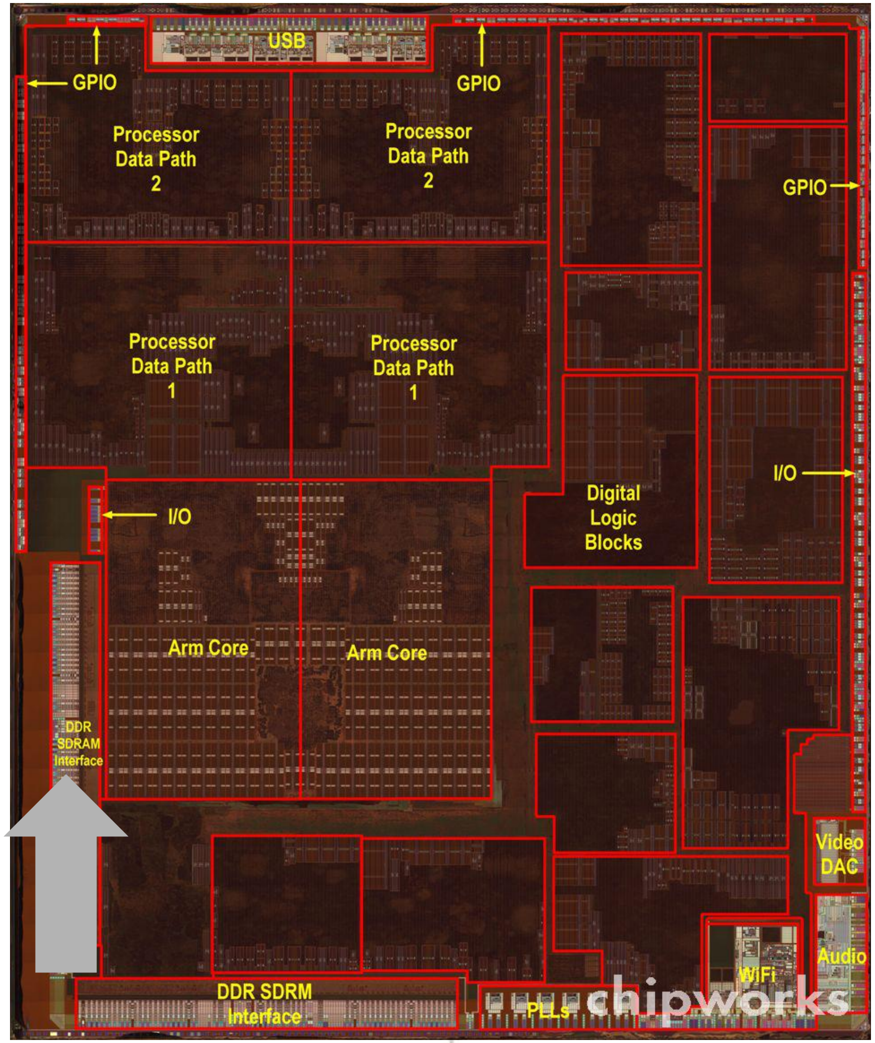
Cortex-A9 MPCore

Floor Plan of A4 and A5

A4

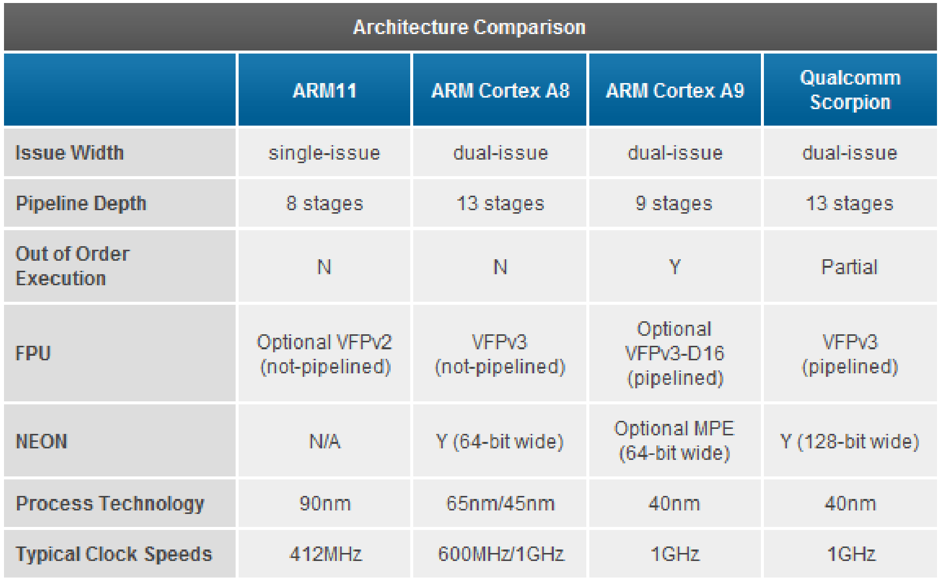


A5

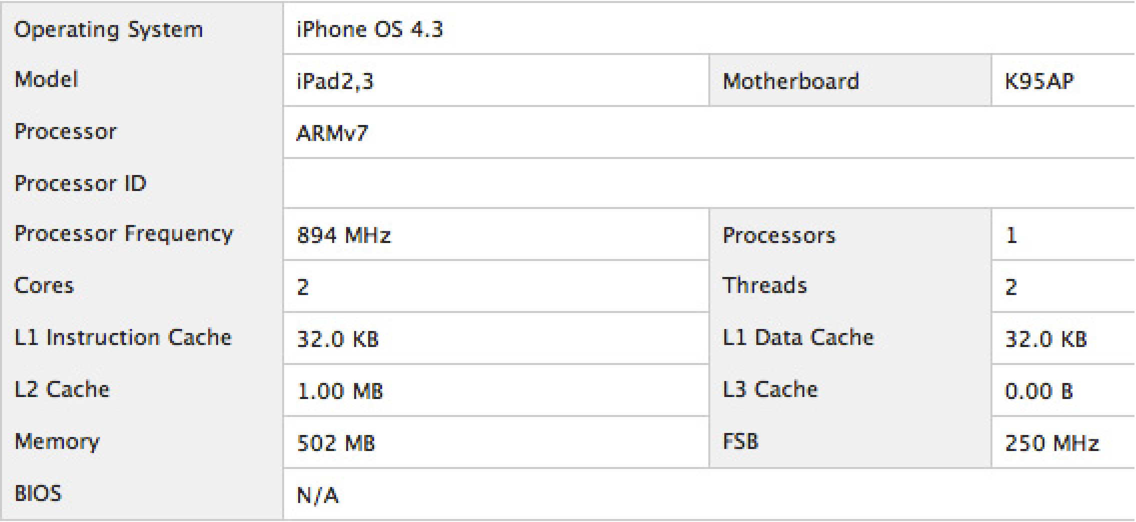


http://www.chipworks.com/en/technical-competitive-analysis/resources/technology-blog/2011/03/apple-a5-vs-a4-floorplan-comparison/

Spec

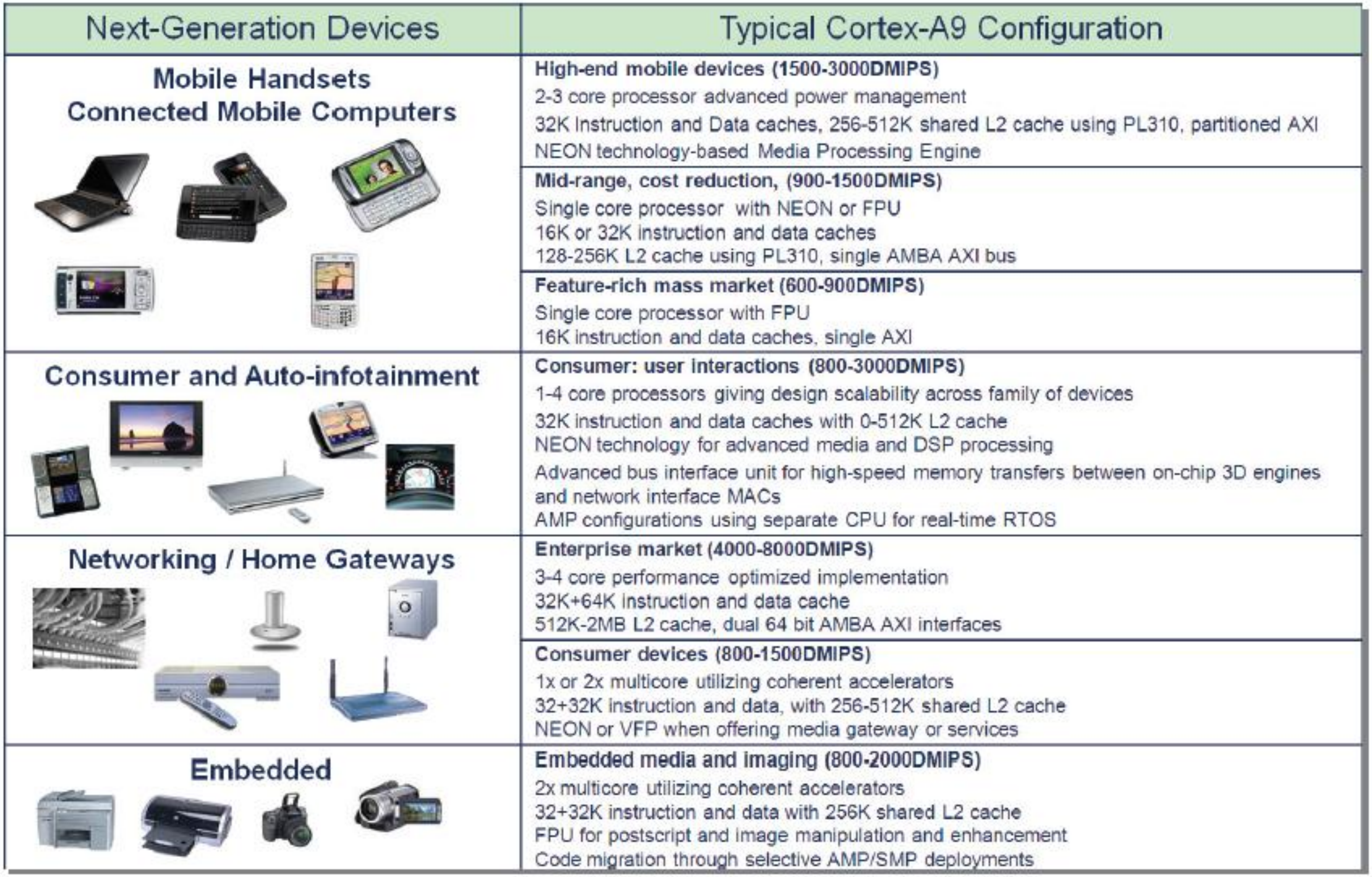


Architecture Comparison				
	ARM11	ARM Cortex A8	ARM Cortex A9	Qualcomm Scorpion
Issue Width	single-issue	dual-issue	dual-issue	dual-issue
Pipeline Depth	8 stages	13 stages	9 stages	13 stages
Out of Order Execution	N	N	Y	Partial
FPU	Optional VFPv2 (not-pipelined)	VFPv3 (not-pipelined)	Optional VFPv3-D16 (pipelined)	VFPv3 (pipelined)
NEON	N/A	Y (64-bit wide)	Optional MPE (64-bit wide)	Y (128-bit wide)
Process Technology	90nm	65nm/45nm	40nm	40nm
Typical Clock Speeds	412MHz	600MHz/1GHz	1GHz	1GHz



Operating System	iPhone OS 4.3		
Model	iPad2,3	Motherboard	K95AP
Processor	ARMv7		
Processor ID			
Processor Frequency	894 MHz	Processors	1
Cores	2	Threads	2
L1 Instruction Cache	32.0 KB	L1 Data Cache	32.0 KB
L2 Cache	1.00 MB	L3 Cache	0.00 B
Memory	502 MB	FSB	250 MHz
BIOS	N/A		

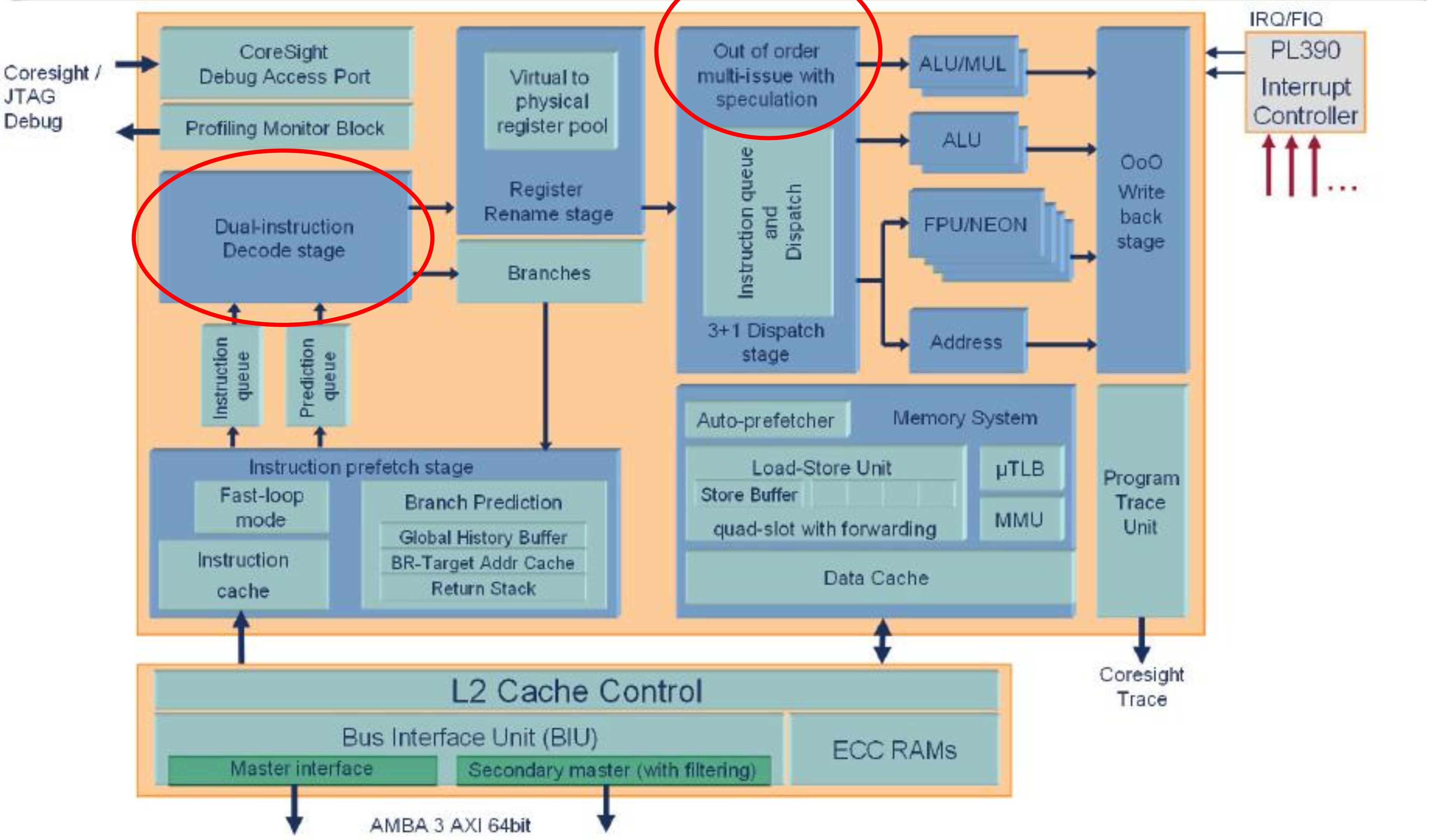
Examples of Using Cortex-A9



Next-Generation Devices	Typical Cortex-A9 Configuration
Mobile Handsets **Connected Mobile Computers**	**High-end mobile devices (1500-3000DMIPS)** 2-3 core processor advanced power management 32K Instruction and Data caches, 256-512K shared L2 cache using PL310, partitioned AXI NEON technology-based Media Processing Engine
	Mid-range, cost reduction, (900-1500DMIPS) Single core processor with NEON or FPU 16K or 32K instruction and data caches 128-256K L2 cache using PL310, single AMBA AXI bus
	Feature-rich mass market (600-900DMIPS) Single core processor with FPU 16K instruction and data caches, single AXI
Consumer and Auto-infotainment	**Consumer: user interactions (800-3000DMIPS)** 1-4 core processors giving design scalability across family of devices 32K instruction and data caches with 0-512K L2 cache NEON technology for advanced media and DSP processing Advanced bus interface unit for high-speed memory transfers between on-chip 3D engines and network interface MACs AMP configurations using separate CPU for real-time RTOS
Networking / Home Gateways	**Enterprise market (4000-8000DMIPS)** 3-4 core performance optimized implementation 32K+64K instruction and data cache 512K-2MB L2 cache, dual 64 bit AMBA AXI interfaces
	Consumer devices (800-1500DMIPS) 1x or 2x multicore utilizing coherent accelerators 32+32K instruction and data, with 256-512K shared L2 cache NEON or VFP when offering media gateway or services
Embedded	**Embedded media and imaging (800-2000DMIPS)** 2x multicore utilizing coherent accelerators 32+32K instruction and data with 256K shared L2 cache FPU for postscript and image manipulation and enhancement Code migration through selective AMP/SMP deployments

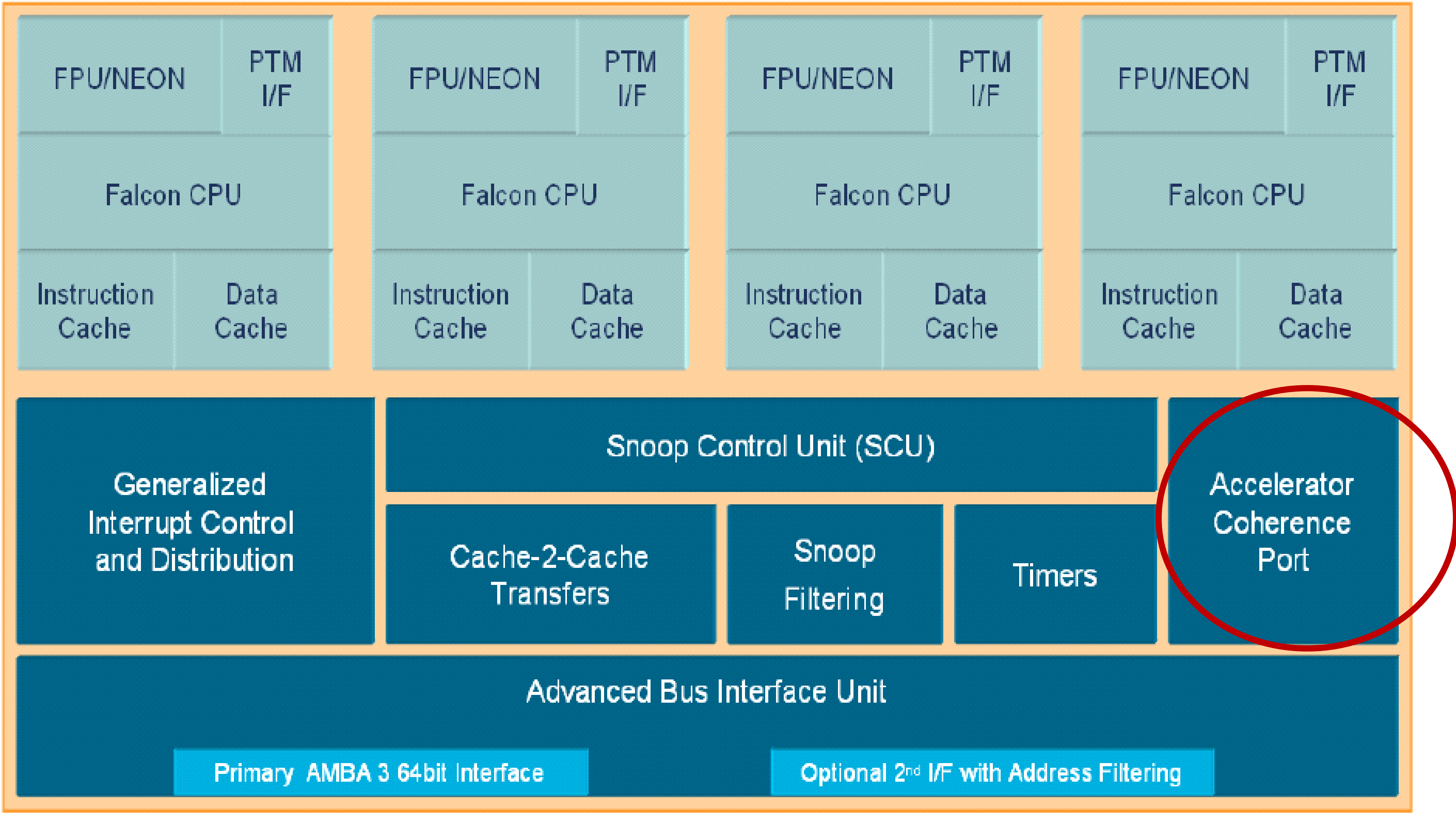
http://www.arm.com/files/pdf/ARMCortexA-9Processors.pdf

Coretex-A9 Micro-Architecture



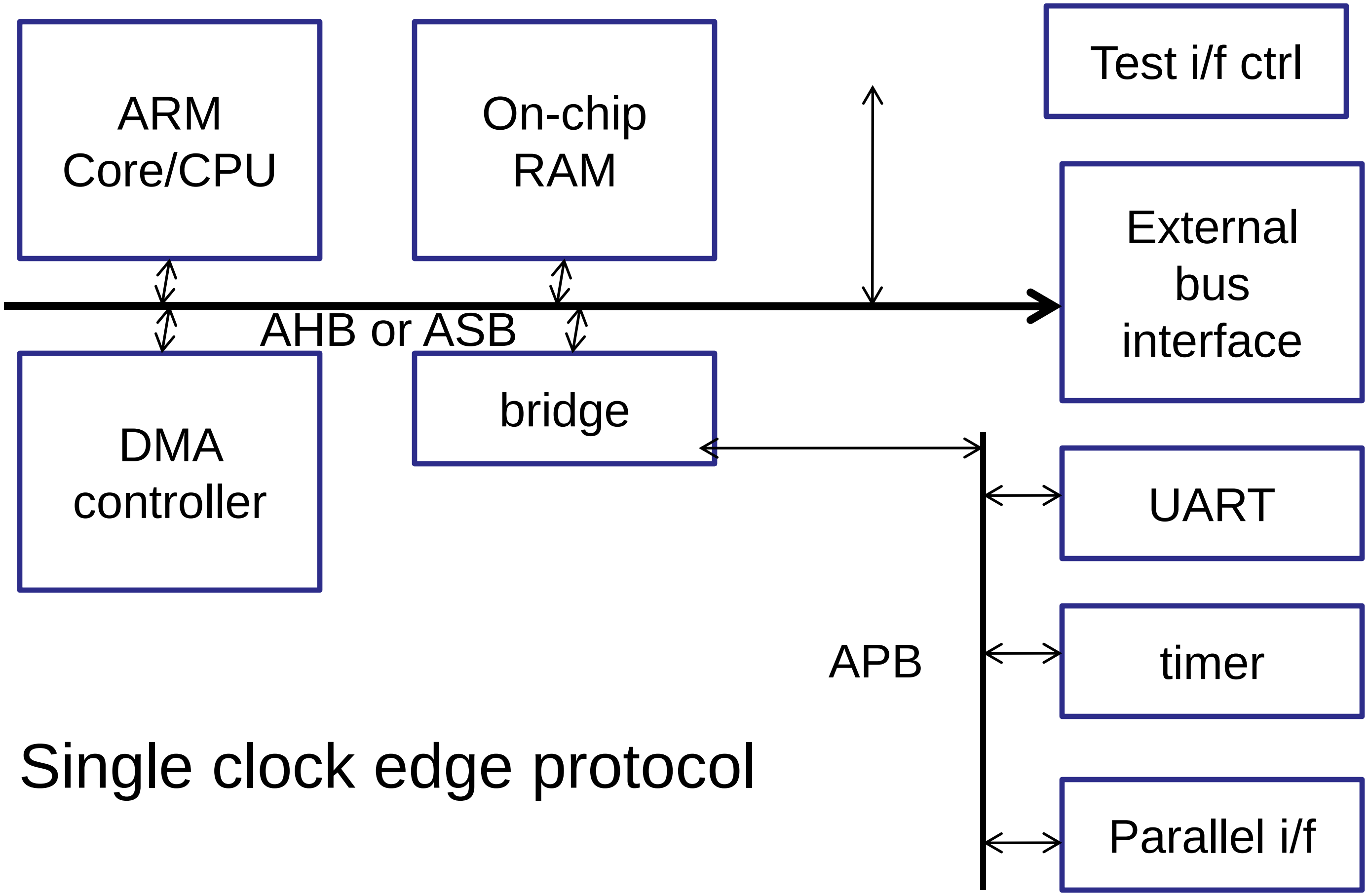
http://www.arm.com/files/pdf/ARMCortexA-9Processors.pdf

MP Cortex A-9 Processors



ARM BUS

Advanced Microcontroller Bus Architecture (AMBA)

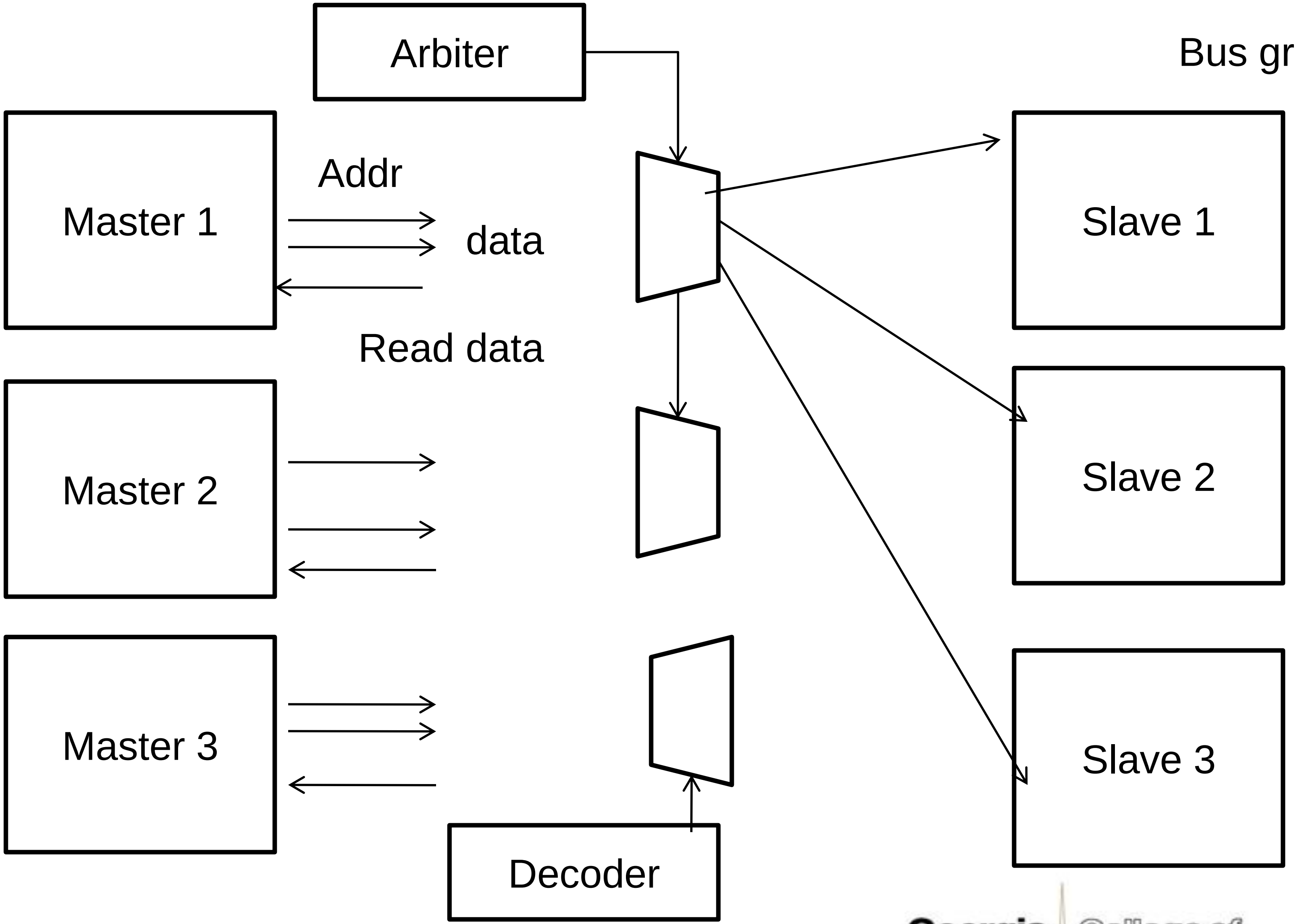


- Single clock edge protocol

AMBA

- AHB (Advanced High-performance Bus)
  - New standard
  - Connect high-performance system
  - Burst mode data transfer and split transactions
  - Pipelined
- ASB (Advanced System Bus)
  - Old standard
  - Connect high-performance system
  - Pipelined
  - Multiple systems
- APB (Advanced Peripheral Bus)
  - A simpler interface for low-performance peripherals
  - Low power
  - Latched address, simple interface

Bus Arbitration



AMBA Arbitration

- A bus transaction is initiated by a bus master which requests access from a central arbiter.
- The arbiter decides priorities when there are conflicting requests.
- The design of the arbiter is a system specific issue.
- The ASB only specifies the protocol:
  - The master issues a request to the arbiter
  - When the bus is available, the arbiter issues a grant to the master.

wuzwuz.nuigalway.ie/.../SOC_Lecture_02_UCG%20November%2012%202002.ppt

Bus Pipelining

- A memory access consists of several cycles (including arbitration)
- Since the bus is not used in all cycles, pipelining can be used to increase performance

Write Access

	AR	ARB	AG	RQ	ACK
Arb request	▨				
Arbiter		▨			
Arb grant			▨		
Bus				▨	▨

Read Access

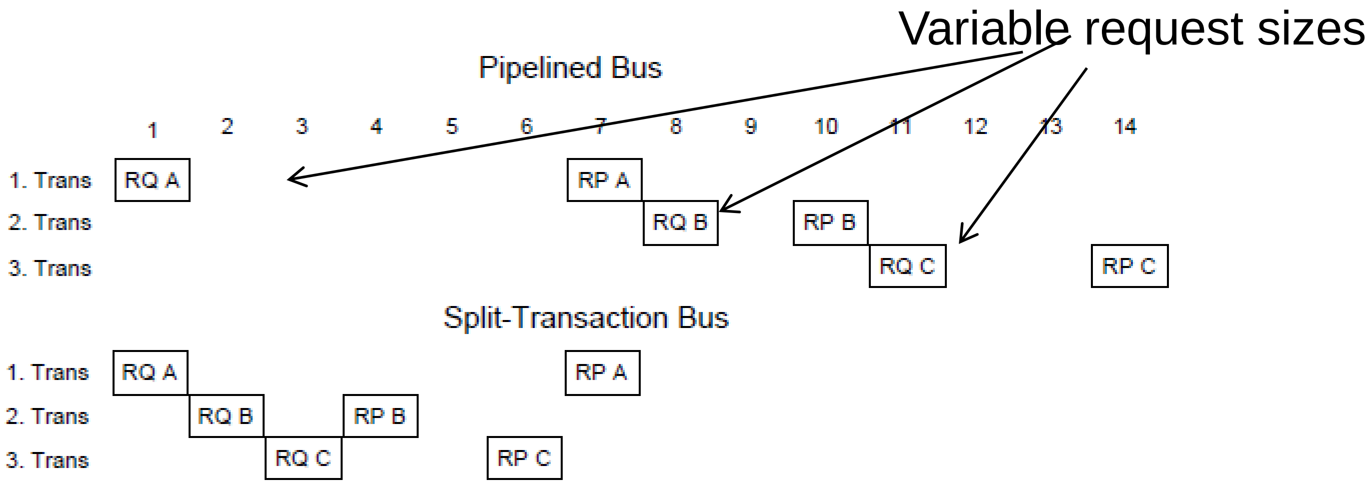
	AR	ARB	AG	RQ	P	RPLY
Arb request	▨					
Arbiter		▨				
Arb grant			▨			
Bus				▨		▨

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1. Read	AR	ARB	AG	RQ	P	RPLY									
2. Write		AR	ARB	AG	Stall	Stall	RQ	ACK							
3. Write			AR	ARB	Stall	Stall	AG	Stall	RQ	ACK					
4. Read				AR	Stall	Stall	ARB	Stall	AG	Stall	RQ	P	RPLY	RQ	
5. Read							AR	Stall	ARB	Stall	AG	RQ	P	RPLY	
6. Read									AR	Stall	ARB	AG	Stall	Stall	RQ
Bus busy				▨		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

http://www.imit.kth.se/courses/2B1447/Lectures/2B1447_L4_Buses.pdf

Split Transactions

- A transaction is splitted into a two transactions
  - Request-transaction
  - Reply-transaction
- Both transactions have to compete for the bus by arbitration

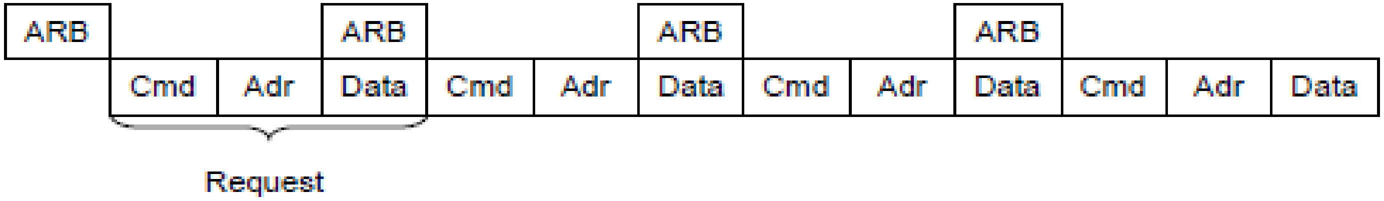


Variable request sizes

http://www.imit.kth.se/courses/2B1447/Lectures/2B1447_L4_Buses.pdf

Burst Messages

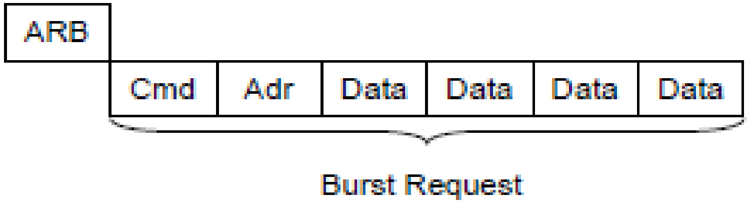
- Overheads can be reduced if the requests are sent as a burst
- Overheads
  - Arbitration, Addressing, Acknowledgement
- Better efficiency, but be careful with long requests



ARB | ARB | ARB | ARB

Cmd | Adr | Data | Cmd | Adr | Data | Cmd | Adr | Data | Cmd | Adr | Data

Request



ARB

Cmd | Adr | Data | Data | Data | Data

Burst Request

http://www.imit.kth.se/courses/2B1447/Lectures/2B1447_L4_Buses.pdf

Bus Bridges

- Bus bridges are used to separate high-performance devices from low-performance devices
- All communication from high-performance bus with the low performance device goes via the bridge

http://www.imit.kth.se/courses/2B1447/Lectures/2B1447_L4_Buses.pdf