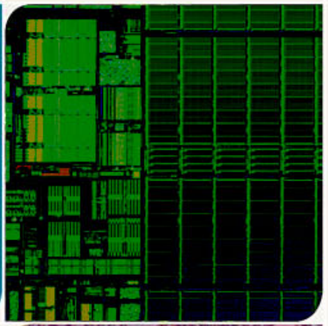
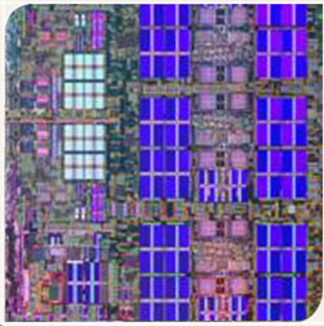
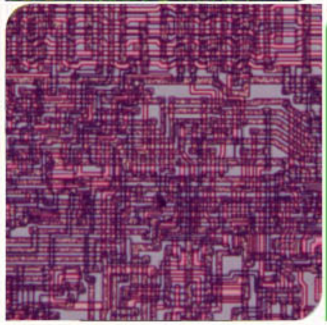
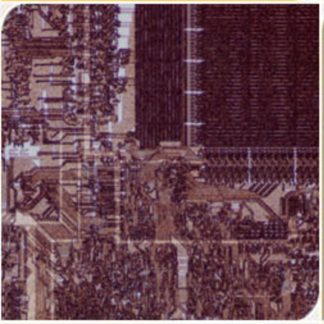





CS4803PGC Design and Programming of Game Console

Spring 2012

Prof. Hyesoon Kim

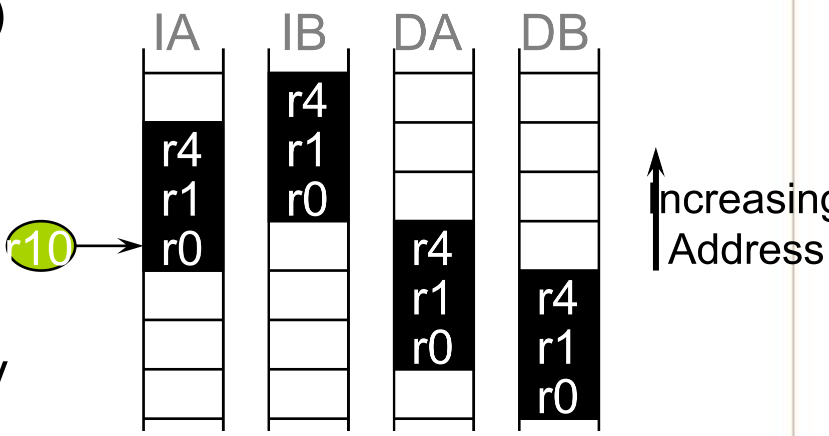


Georgia Tech College of Computing

Full/Empty Ascending/Descending Stacks

- Descending (address grows download)
- Ascending (Address grows upward)

- Full/Empty: the stack pointer can be either point to the last item (a full stack) or the next free space (an empty stack)



Stack Type	Push	Pop
Full descending	STMFD /	LDMFD /
Full Ascending	STMFA /	LDMFA /
Empty Descending	STMED /	LDMED /
Empty Ascending	STMEA /	LDMEA /

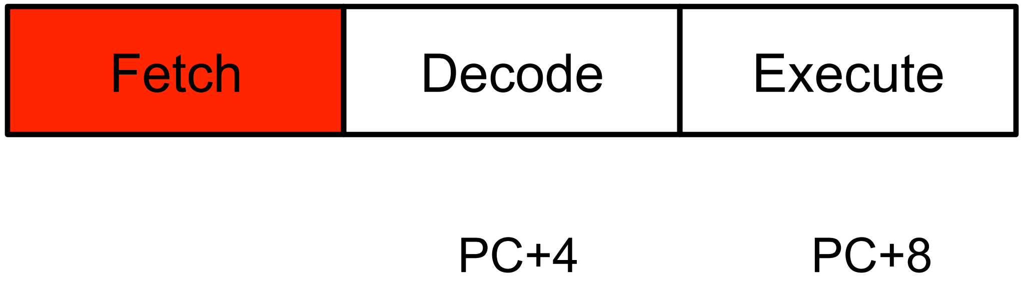
Georgia Tech College of Computing

Use of R15

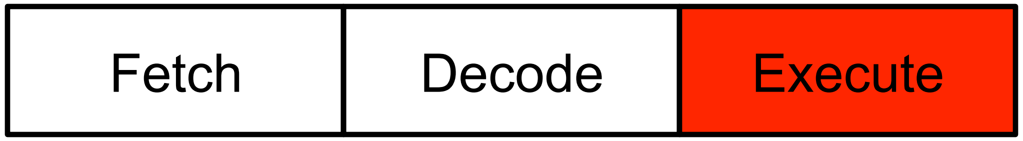
- R15: PC
  - PC may be used as a source operand
  - Register-based shift cannot use R15 as source operands.
- Running-ahead PC's behavior
  - PC is always running ahead
  - PC is always pointing +8 of the current instruction
    - Imagine 3-stage pipeline machine . PC is pointing what to fetch when 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



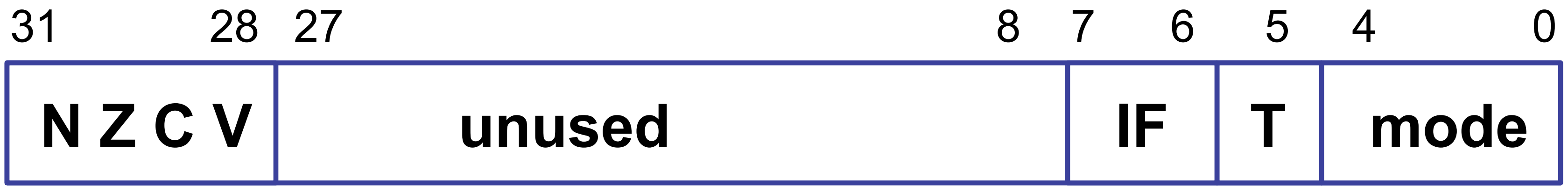
- Data abort : memory execution



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)
```



Bit 7: interrupt
Bit 6: Fast interrupt

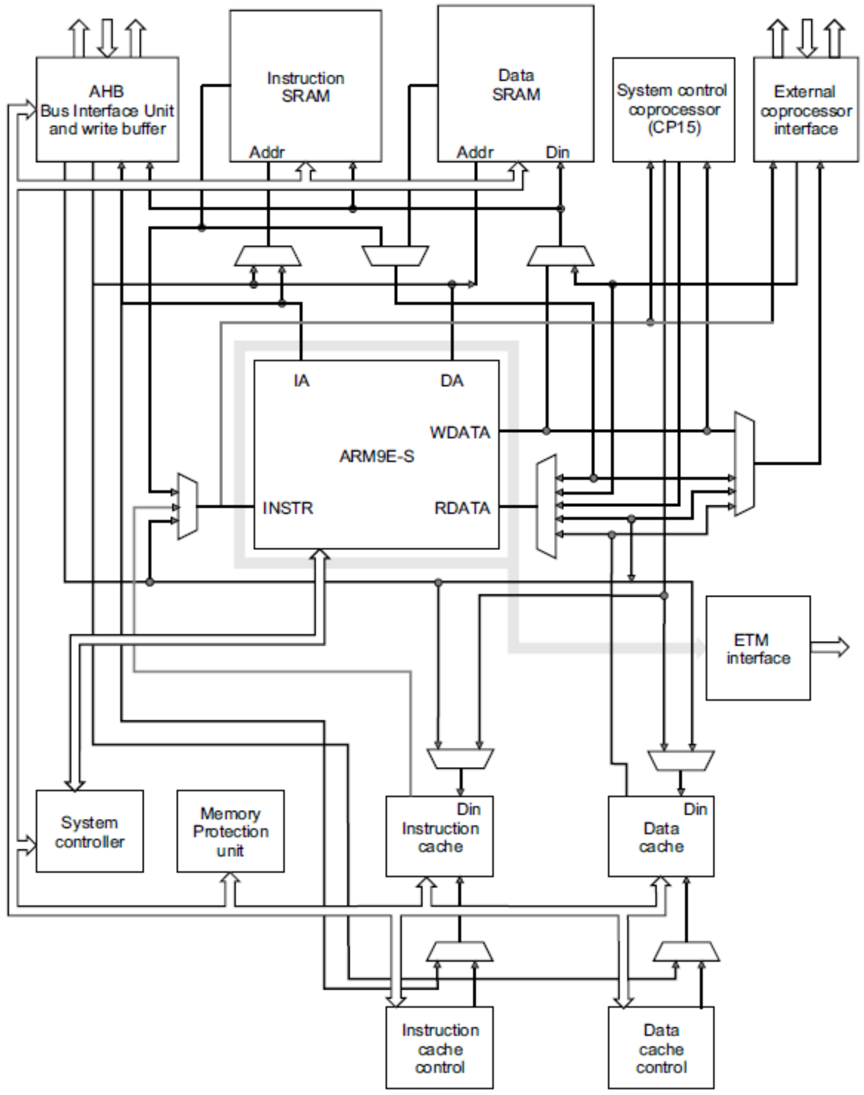
Typical example of interrupt handler

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

```
; ….
LDMFD sp!, {reglist,pc}^
```

ARM946E-S:(ARM 9 in Nintendo DS)

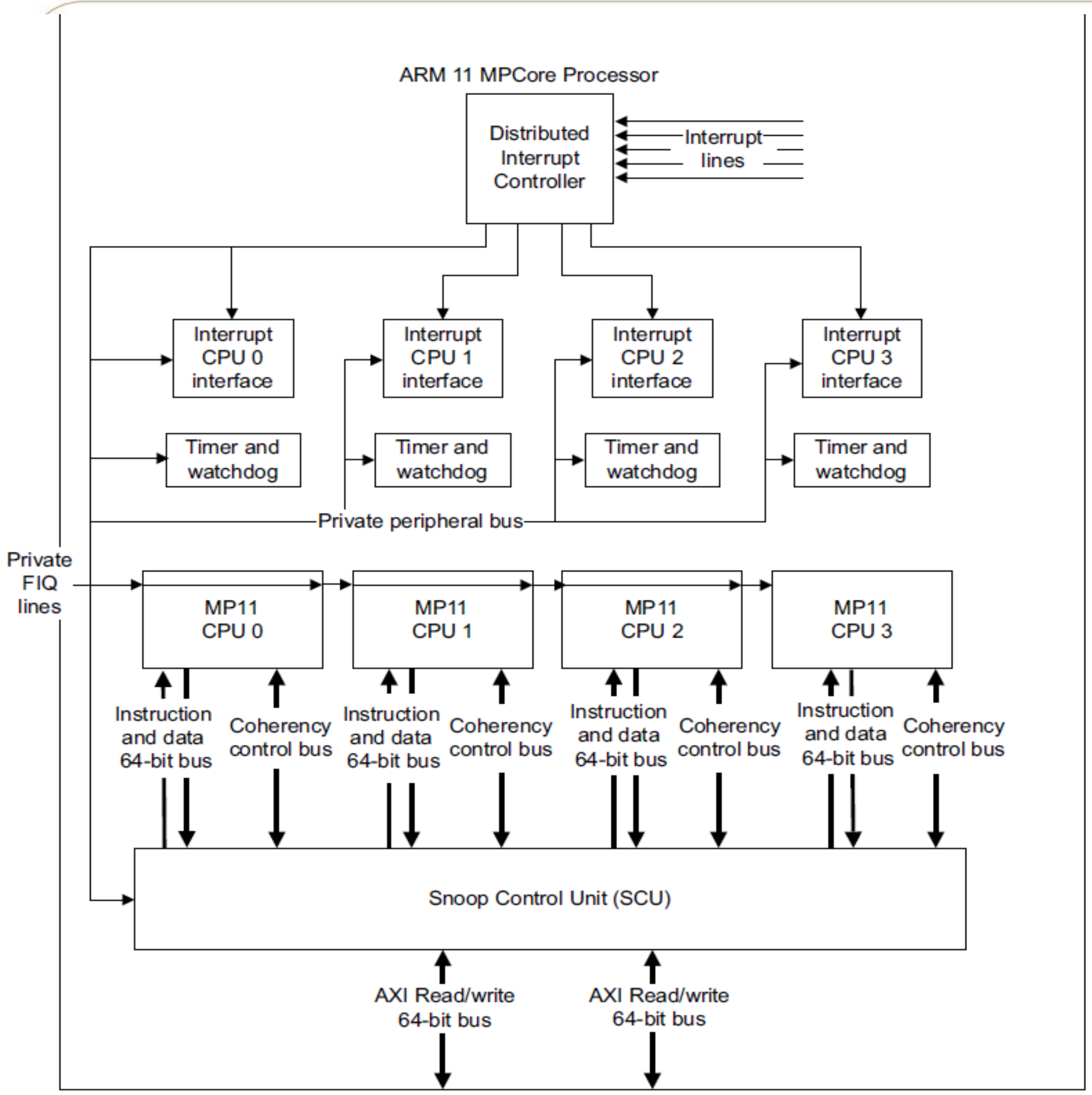
- Soc for embedded system.
- Single chip DSP
- Embedded applications running an RTOS
- Mass storage - HDD & DVD
- Speech coders
- Automotive control
  - Cruise control, ABS, etc.
- Hands-free interfaces
- Modems and soft-modems
- Audio decoding
- Dolby AC3 digital
- MPEG MP3 audio
- Speech recognition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

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

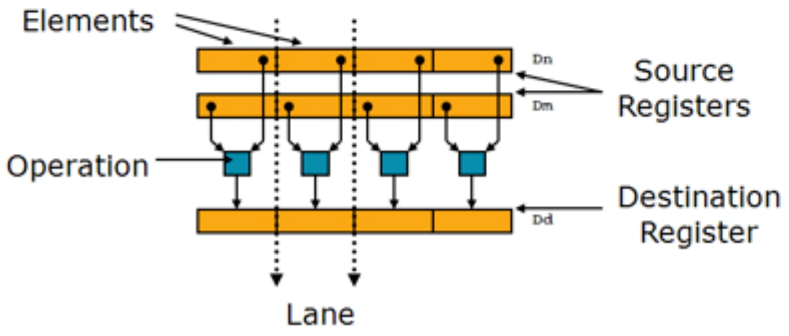
The ARM11 Microarchitecture, David Cormie, ARM Ltd

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

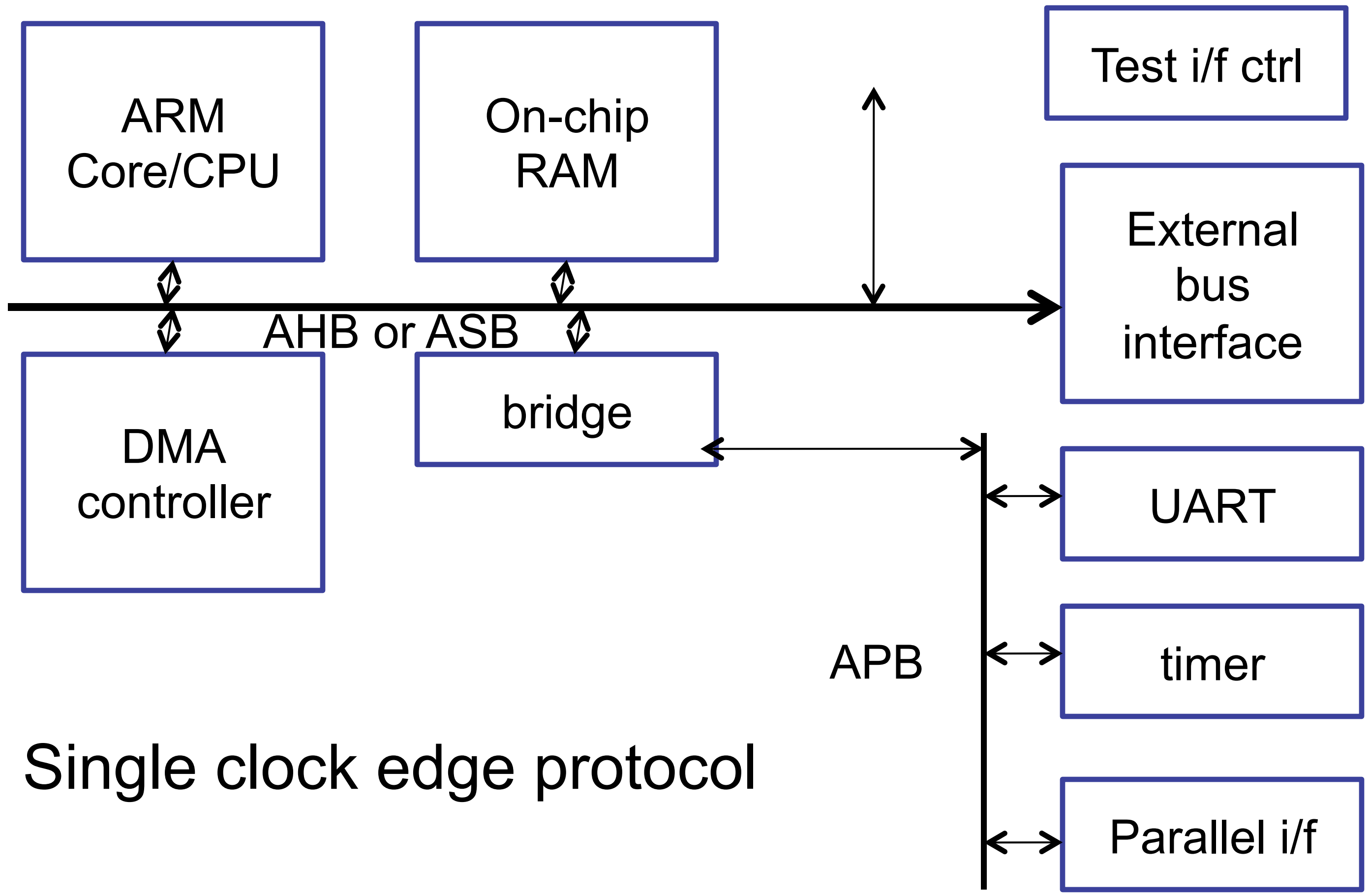
Usage model of NEON

- 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

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

AMBA Revisions

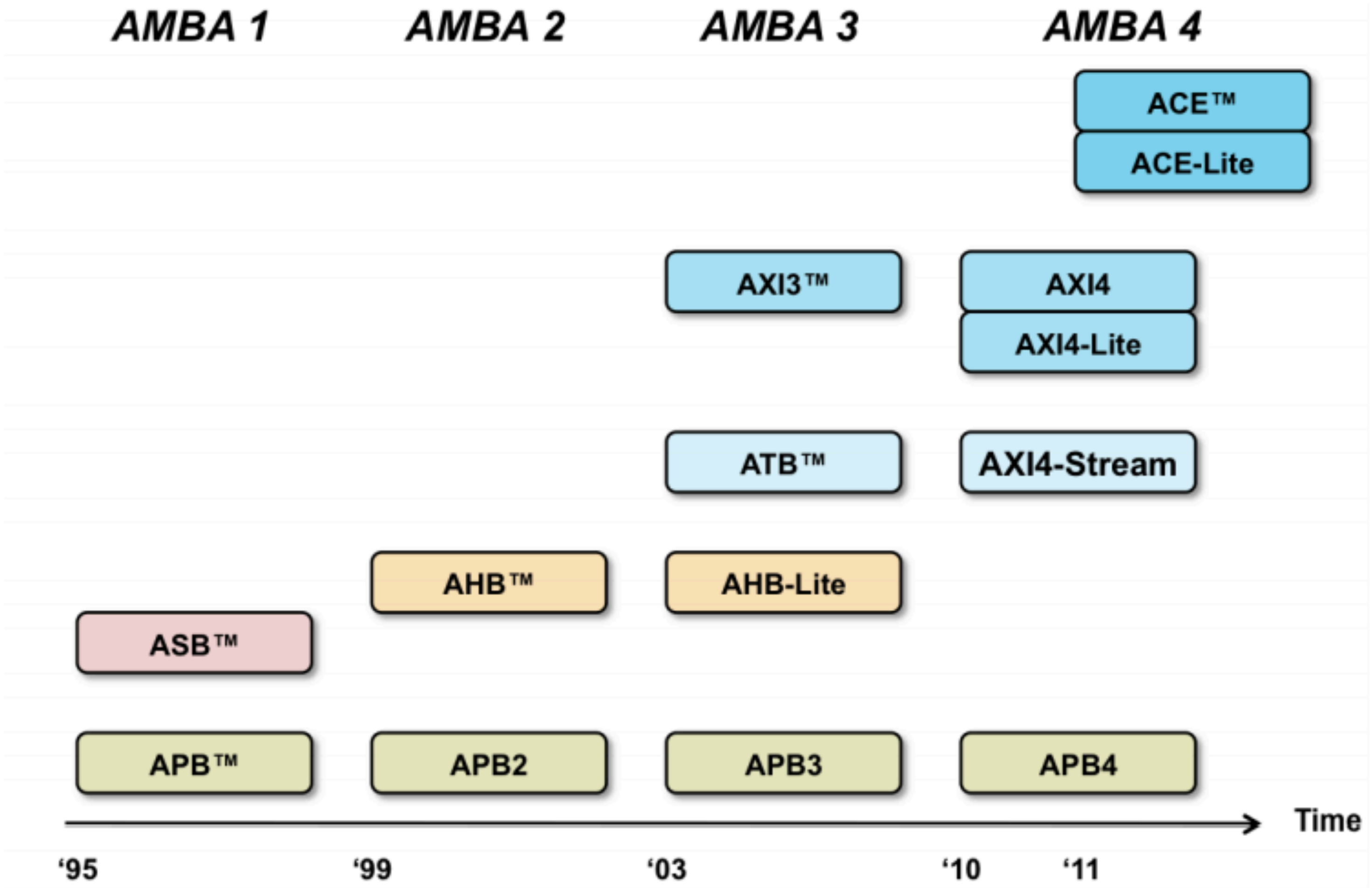
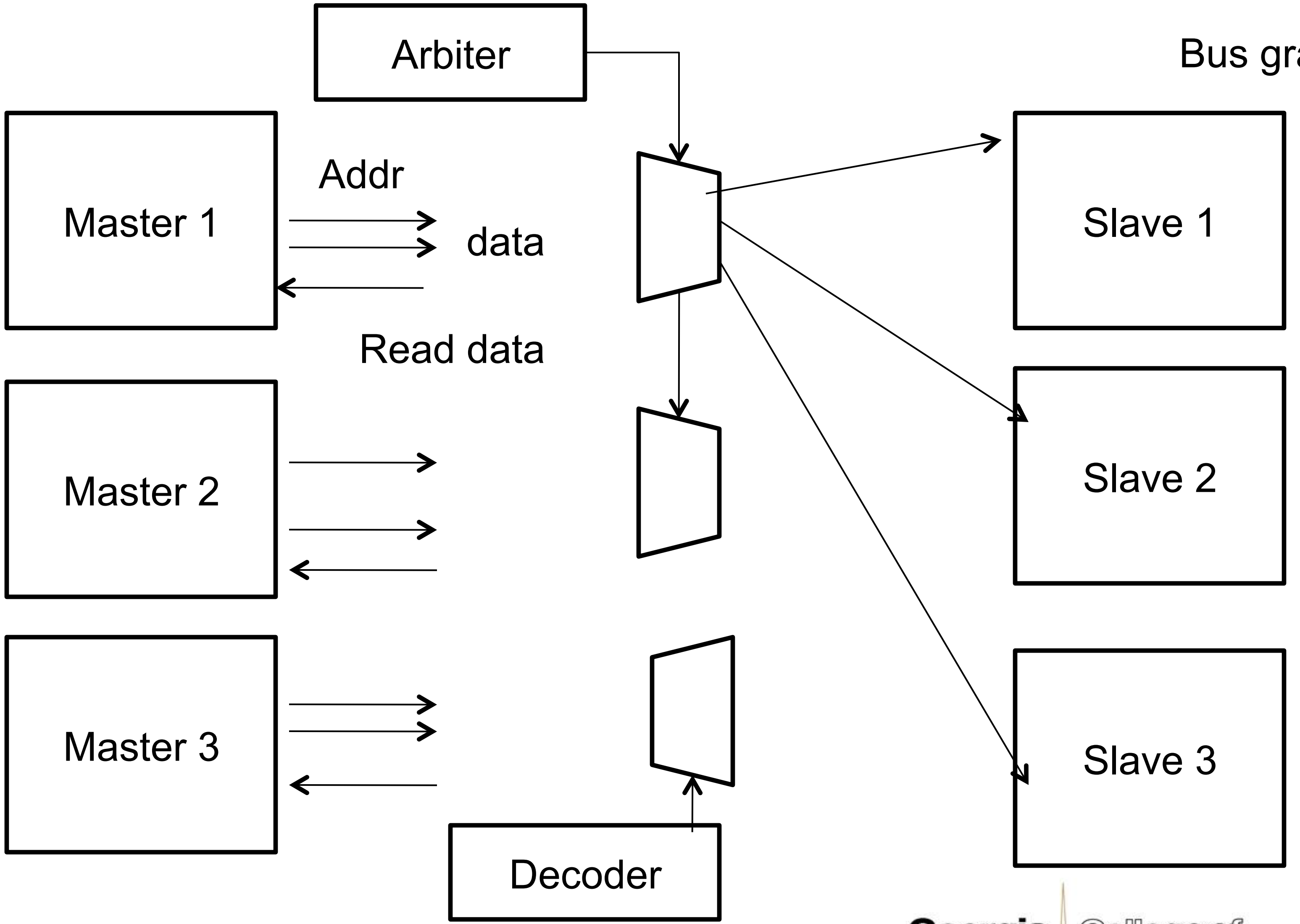


Figure 1 – Evolution of AMBA Standards

http://www.arm.com/files/pdf/CacheCoherencyWhitepaper_6June2011.pdf

Bus Arbitration

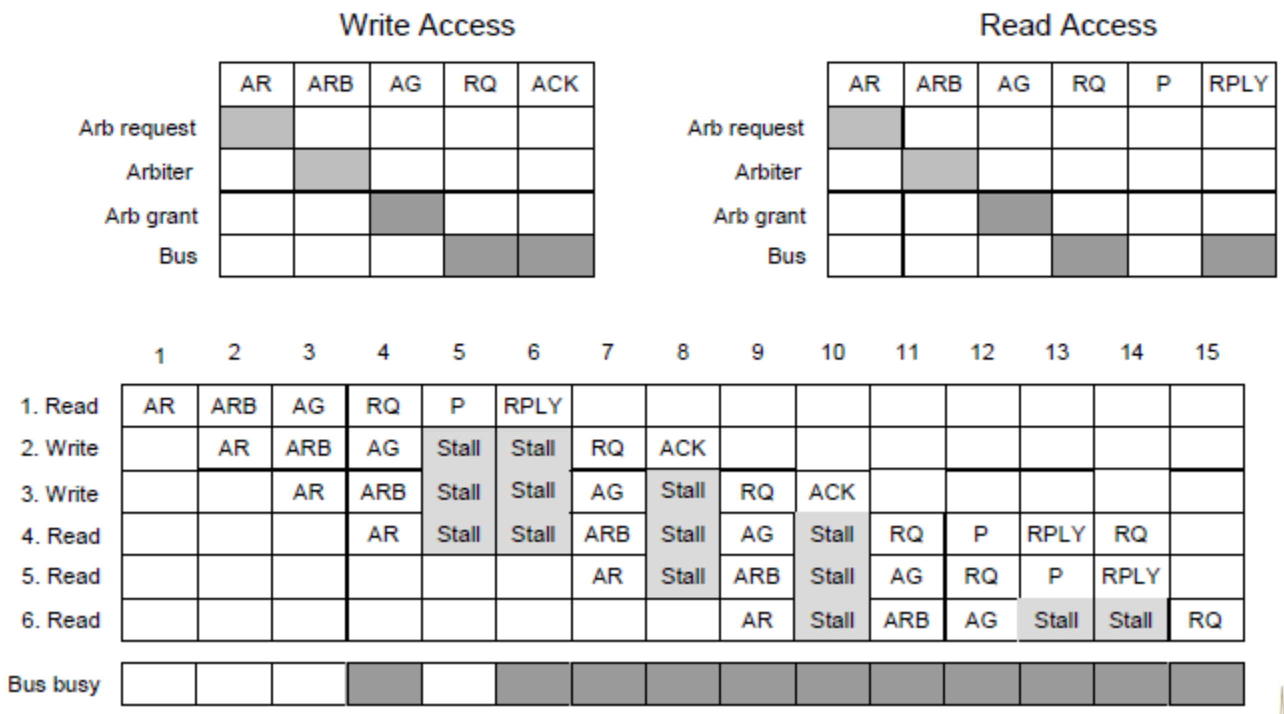


AMBA Arbitration

- 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.

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				▨	▨

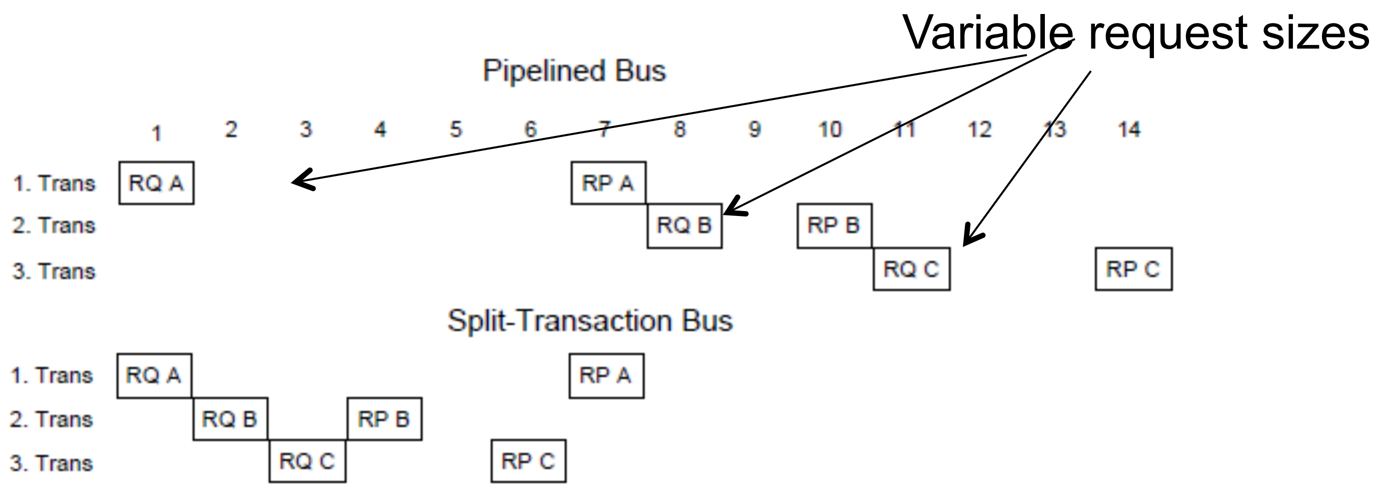
Read Access

	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				▨		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

Split Transactions

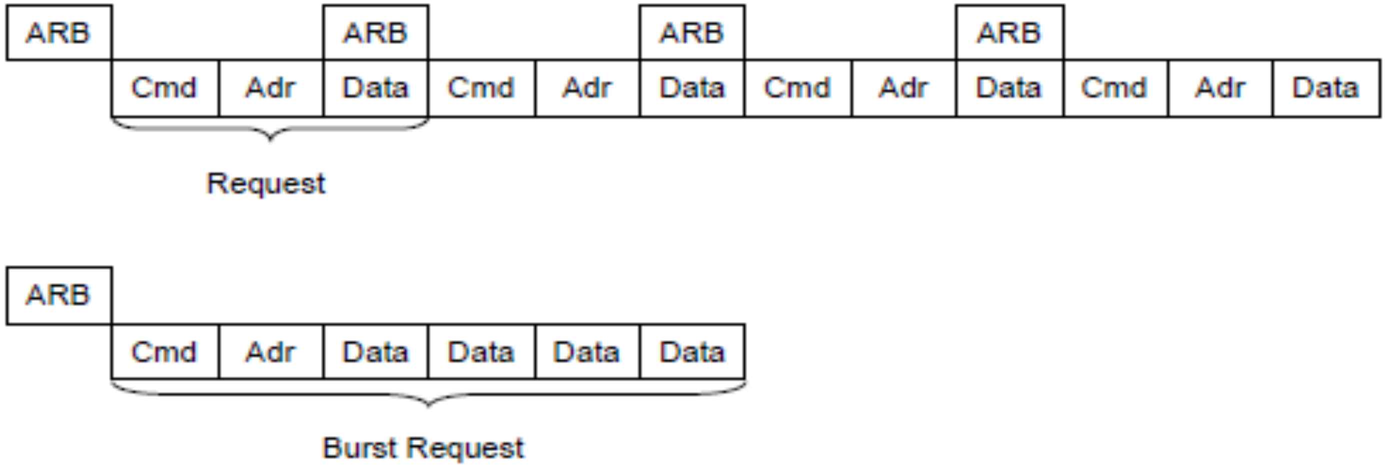
- A transaction is splitted into a two transactions
  - Request-transaction
  - Reply-transaction
- Both transactions have to compete for the bus by arbitration



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



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