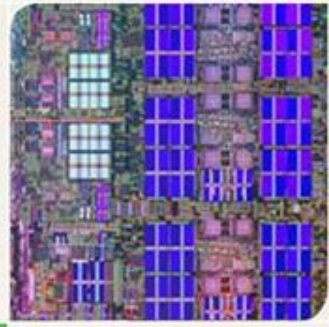
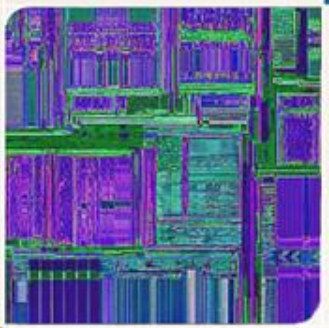
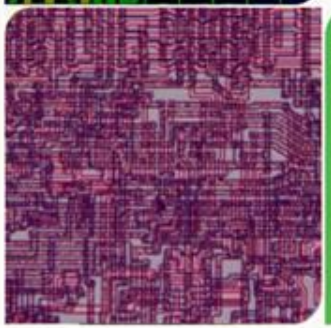



CS4803DGC Design Game Console

Spring 2010

Prof. Hyesoon Kim

Georgia Tech College of Computing

ARM



Brief History of ARM

- ARM is short for Advanced Risc Machines Ltd.
  - Founded 1990, owned by Acorn, Apple and VLSI
- Known before becoming ARM as computer manufacturer
- ARM is one of the most licensed company
- Used especially in portable devices due to low power consumption and reasonable performance (MIPS/watt)
- They do not fabricate silicon

http://tisu.it.jyu.fi/embedded/TIE345/luentokalvot/Embedded_6_ARM.pdf



ARM ISA

- 32-bit wide (16-bit thumb compressed format)
- Load-store instruction set architecture
- 3-address data processing instructions
- Conditional execution of every instruction
- Powerful load and store multiple register instructions
- A general shift operation and a sequential ALU operations in a single instruction that executes in a single clock cycle
- Open instruction set extension through the coprocessor instruction set, including adding new registers and data types to the programmer's model
- Compressed 16-bit thumb architecture

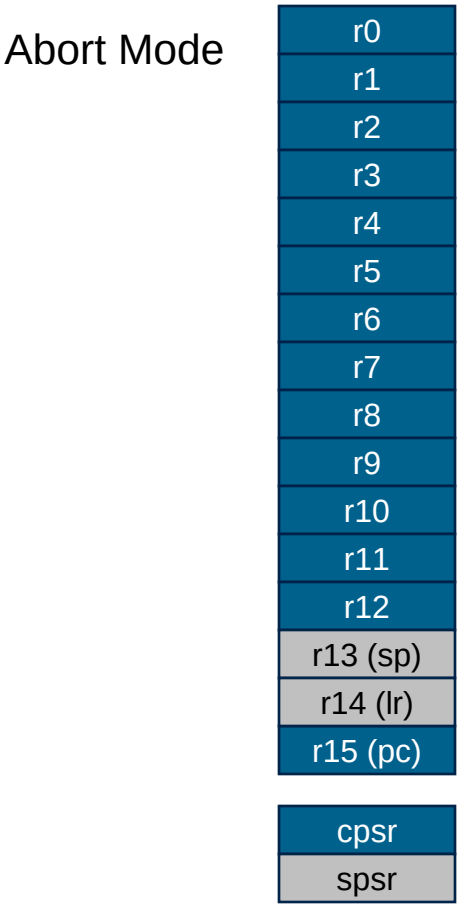


Load-store architecture

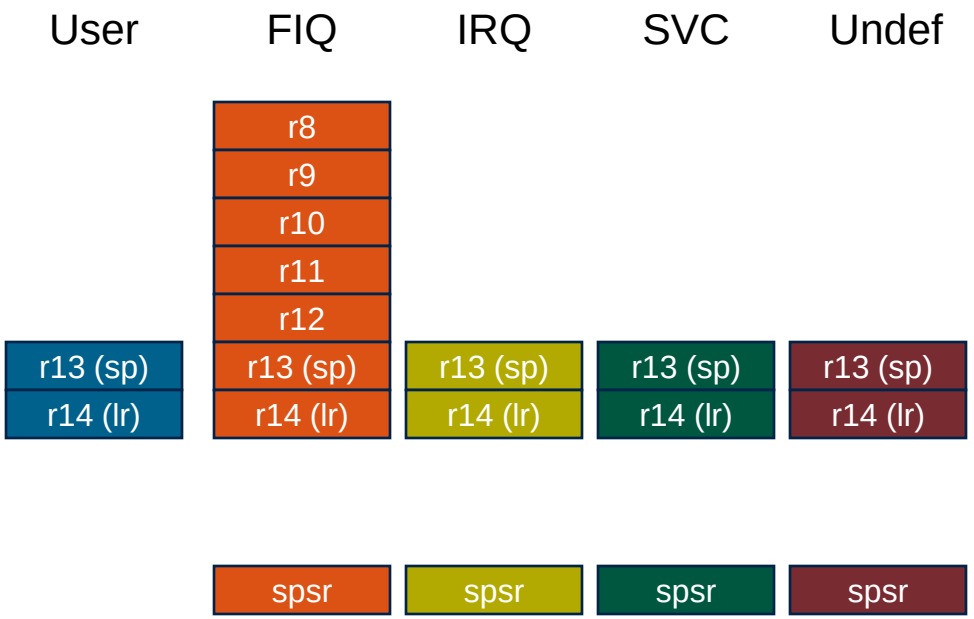
- Data processing (ALU) operations write results only into registers
- Memory operations are only copy (from memory to registers, register to memory)
- ARM does not support memory-to-memory operations
- ARM instruction three categories
  - 1. data processing instructions
  - 2. Data transfer instructions
    - memory-to/from-registers, exchange-memory-register (system only)
  - 3. Control flow instructions
    - Branch instructions, branch and link register (saving return address), trap instructions (supervisor calls)

ARM Register Set

Usable in user mode



System modes only



Current Program Status Register (CPSR)

- N: Negative (the last ALU operation)
- Z: zero (the last ALU operation)
- C: carry (the last ALU or from shifter)
- V: overflow



ARM Operation Modes

CPSR[4:0]	Mode	Use	Registers
10000	user	Normal user code	user
10001	FIQ	Processing fast interrupts	_fiq
10010	IRQ	Processing standard interrupts	_irq
10011	SVC	Processing software interrupts (SWIs)	_svc
10111	Abort	Processing memory faults	_abt
11011	Undef	Handling undefined instruction traps	_und
11111	System	Running privileged operating system tasks	user

Software interrupt: supervisor calls

Memory System

- A linear array of byte address
- Data format (8-bit bytes, 16-bit half-words, 32-bit words)
- Aligned address accesses
- Little endian

Bit 31 Bit 0

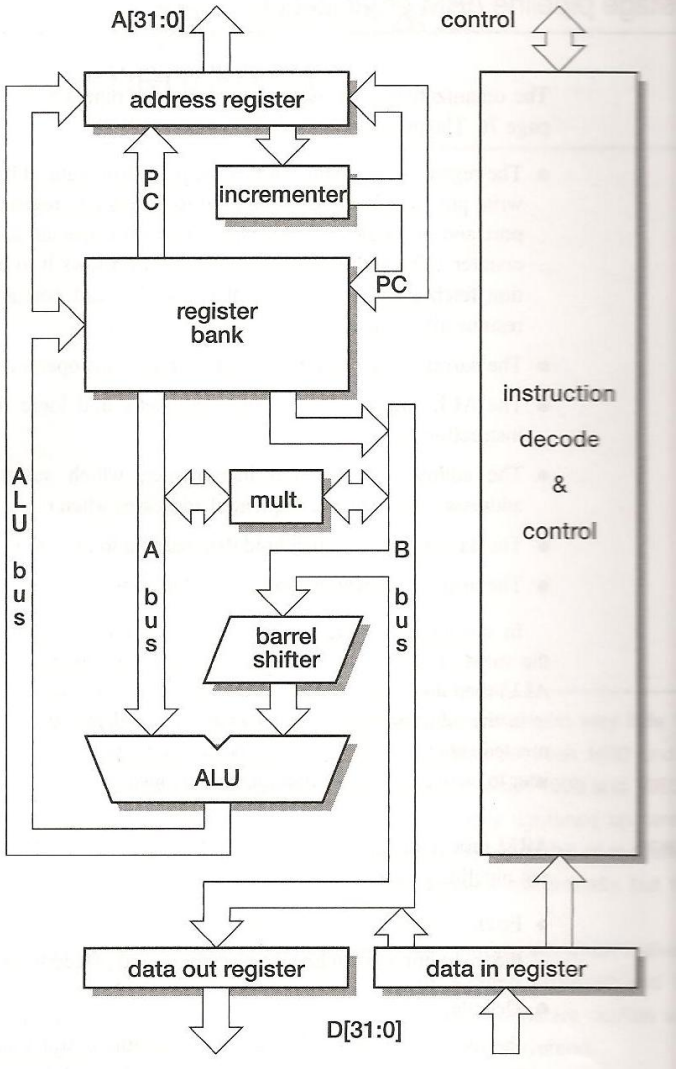
23	22	21	20
19	18	17	16
15	14	13	12
11	10	9	8
7	6	5	4
3	2	1 Byte 1	0 Byte 0



ARM ARCHITECTURE

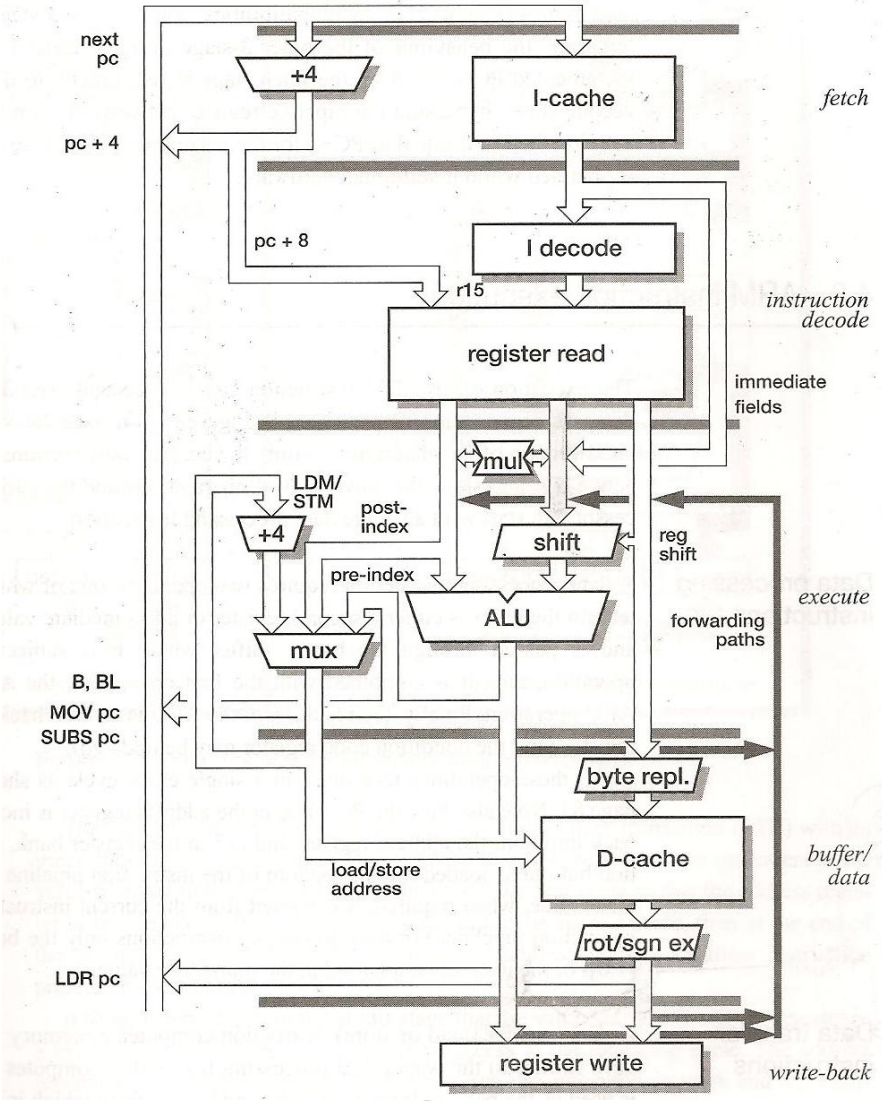
3-stage Pipeline

- Fetch/Decode/Execute
- Allow multi-cycle execution
- Register, two read ports, one write port,
  - Additional register read/write for r15 (program counter)



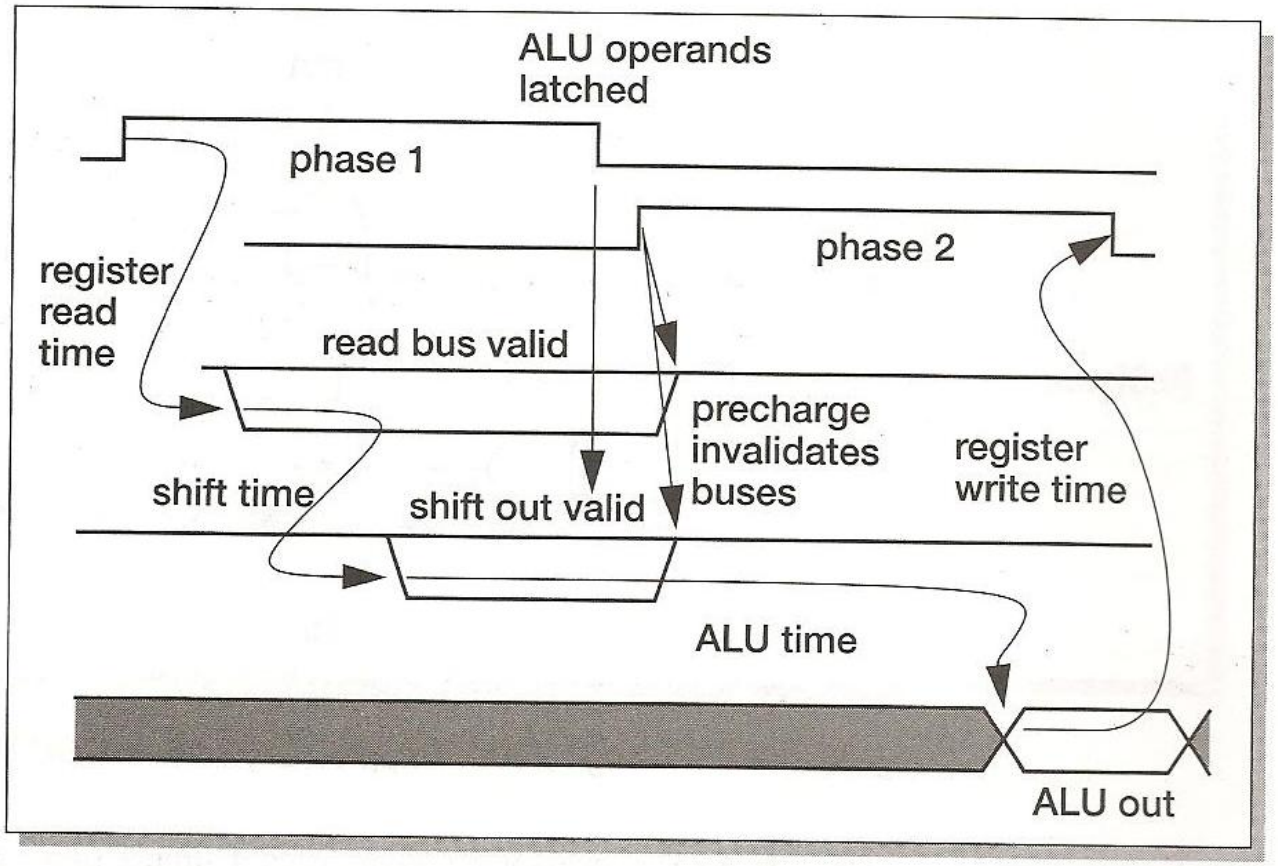
5-Stage Pipeline Processors

- Fetch/Decode/Execute/Mem/write-back
- Introduce forwarding path



ARM Register Timing

- 2-Phase non-overlapping clock scheme





ARM ISA

Privileged Modes

- SPSR (Saved Program Status Register)

M[4:0]	Mode	Accessible register set	
10000	User	PC, R14..R0	CPSR
10001	FIQ	PC, R14_fiq..R8_fiq, R7..R0	CPSR, SPSR_fiq
10010	IRQ	PC, R14_irq..R13_irq, R12..R0	CPSR, SPSR_irq
10011	Supervisor	PC, R14_svc..R13_svc, R12..R0	CPSR, SPSR_svc
10111	Abort	PC, R14_abt..R13_abt, R12..R0	CPSR, SPSR_abt
11011	Undefined	PC, R14_und..R13_und, R12..R0	CPSR, SPSR_und



Thumb Instruction

- 16 bits long
- Similarity with ARM ISA
  - The load-store architecture with data processing, data transfer, and control-flow instructions
  - Support Byte, half-word, word (aligned accesses)
  - A 32-bit unsegmented memory
- Differences
  - Most Thumb instructions are executed unconditionally
    - All ARM instructions are executed conditionally
  - Many thumb data processing instructions use a 2-address format
  - Thumb instruction formats are less regular than ARM ISA.

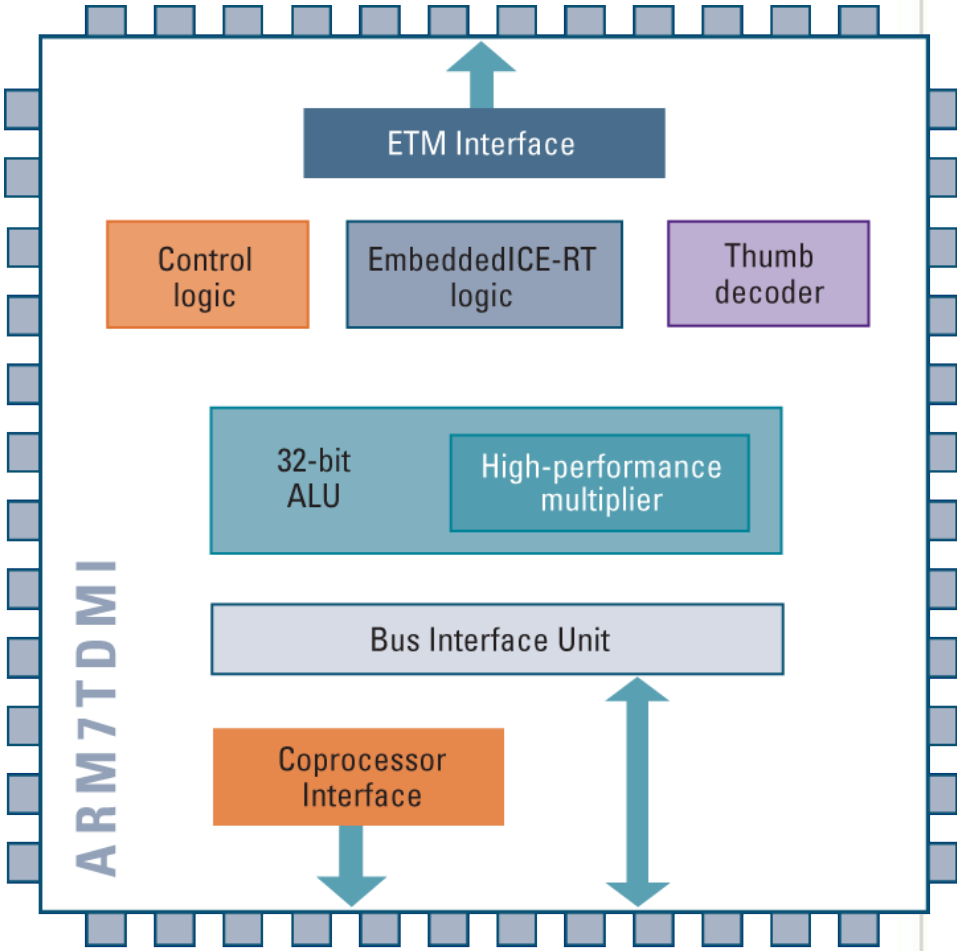


ARM7 and ARM7 TDMI

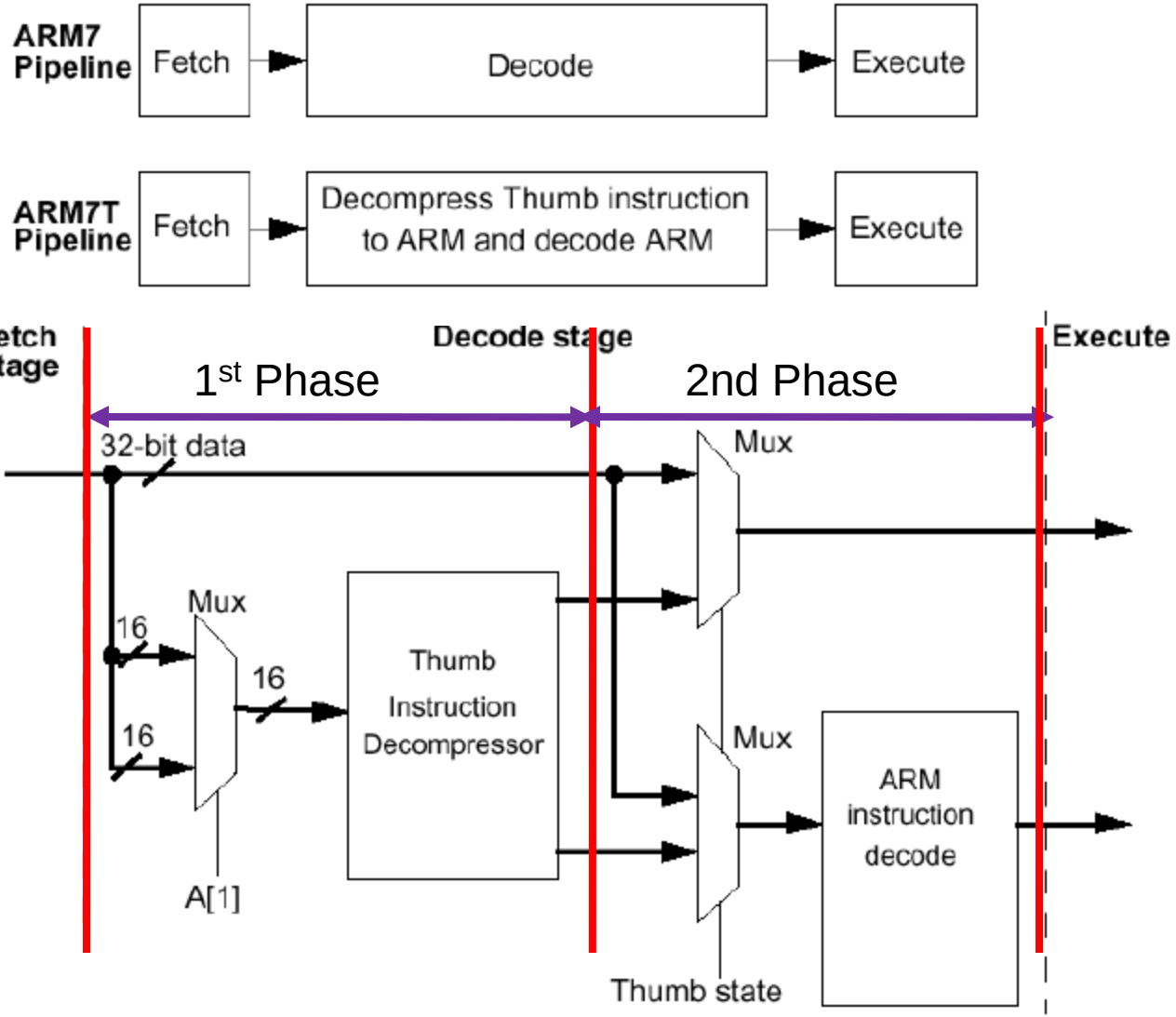
- ARM7: 3 stage pipeline, 16 32-bit Registers , 32-bit instruction set
- TMDI
  - Thumb instruction set
  - Debug-interface
  - Multiplier (hardware)
  - Interrupt (fast interrupt)
  - The most commonly used one

ARM7 TDMI

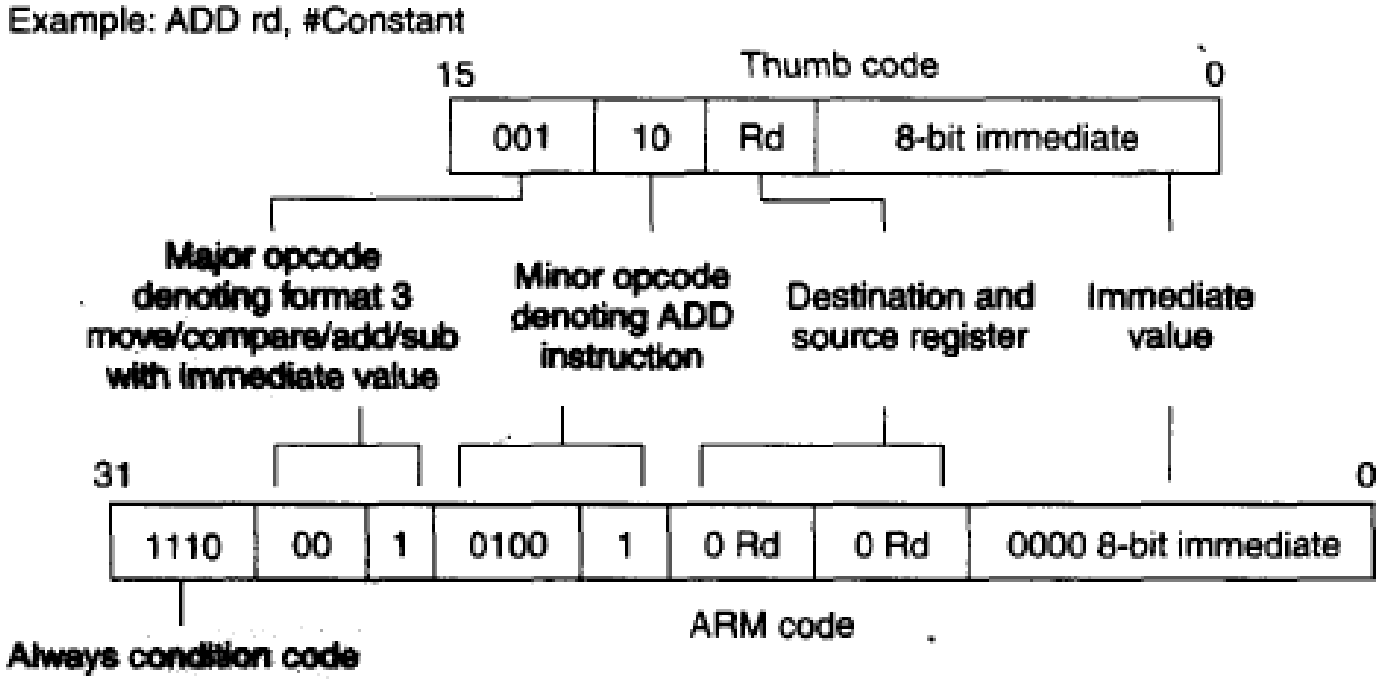
- 32/16-bit RISC
- 32-bit ARM instruction set
- 16-bit Thumb instruction set
- 3-stage pipeline
- Very small die size and low power
- Unified bus interface (32-bit data bus carries both instruction, data)



Thumb Instruction Decode



Thumb Instruction



- Instruction compression to save I-cache/memory accesses
- Use only top 8 registers,
- 3 operands → 2 operands



Thumb…

- Instructions are compiled either native ARM code or Thumb code
  - To utilize full 16bit opcode
  - Use current processor status register (CPSR) to set thumb/native instruction

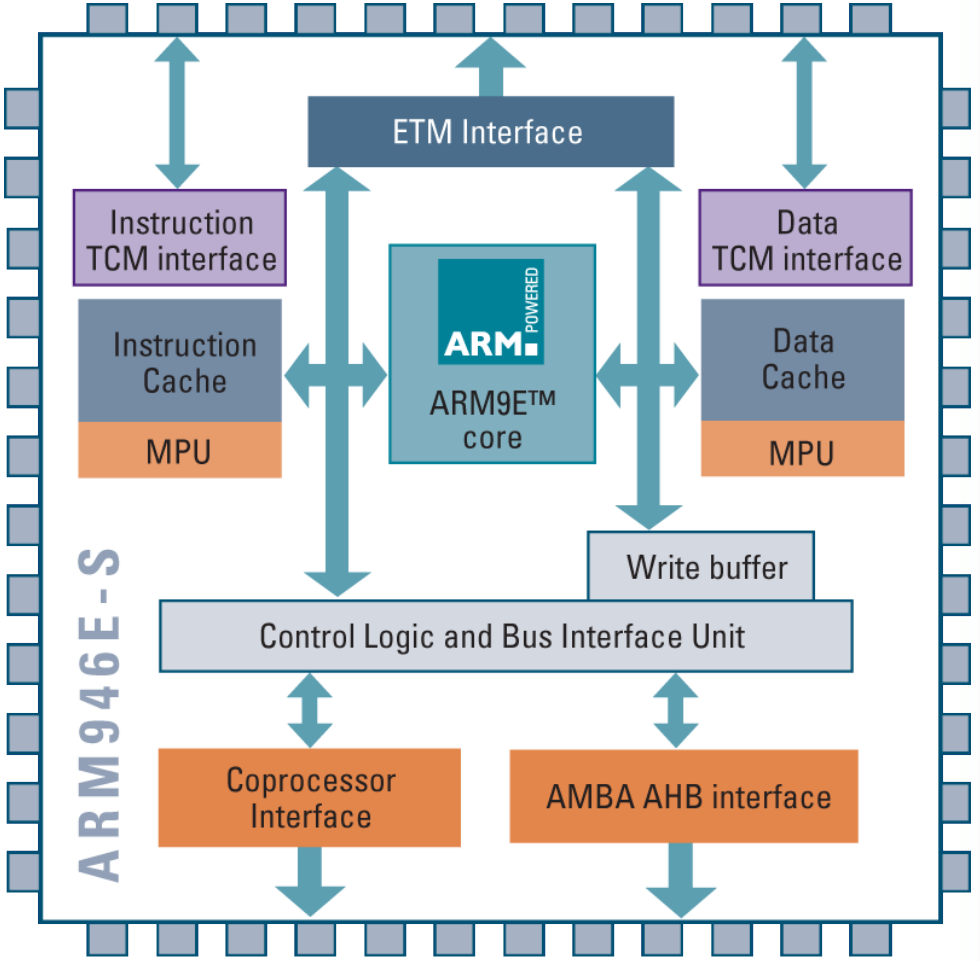


ARM Instruction Set

- All instructions are conditional
- BX, branch and eXhange → branch and exchange (Thumb)
- Link register (subroutine Link register)
  - R14 receives the return address when a Branch with Link (BL or BLX) instruction is executed

ARM9

- 5-stage pipeline
- I-cache and D-cache
- Floating point support with the optional VFP9-S coprocessor
- Enhanced 16 x 32-bit multiplier capable of single cycle MAC operations
- The ARM946E-S processor supports ARM's real-time trace technology



ARM9

- ARM7 3stage->ARM9 5 stage
  - Increase clock frequency

ARM7TDMI Pipeline Operation

Fetch	Decode	Execute
Instruction Fetch	Convert Thumb to ARM Main Decode Register Address Decode	Register Read Shifter ALU Writeback

ARM9TDMI Pipeline Operation

Fetch	Decode	Execute	Memory	Writeback
Instruction Fetch	ARM Decode Reg. Address Decode Register Read Thumb Decode Reg. Address Decode Register Read	Shifter ALU	Memory Data access	ALU Result and / or Load data Writeback

ARM9 Pipeline

- ARM7: Thumb instruction decode: first ½ phase of decode stage
- ARM9: Parallel decoding
- ARM7: ALU (arithmetic, and logic units) is active all the time
- ARM9: Two units are partitioned to save power
- ARM9: Forwarding path