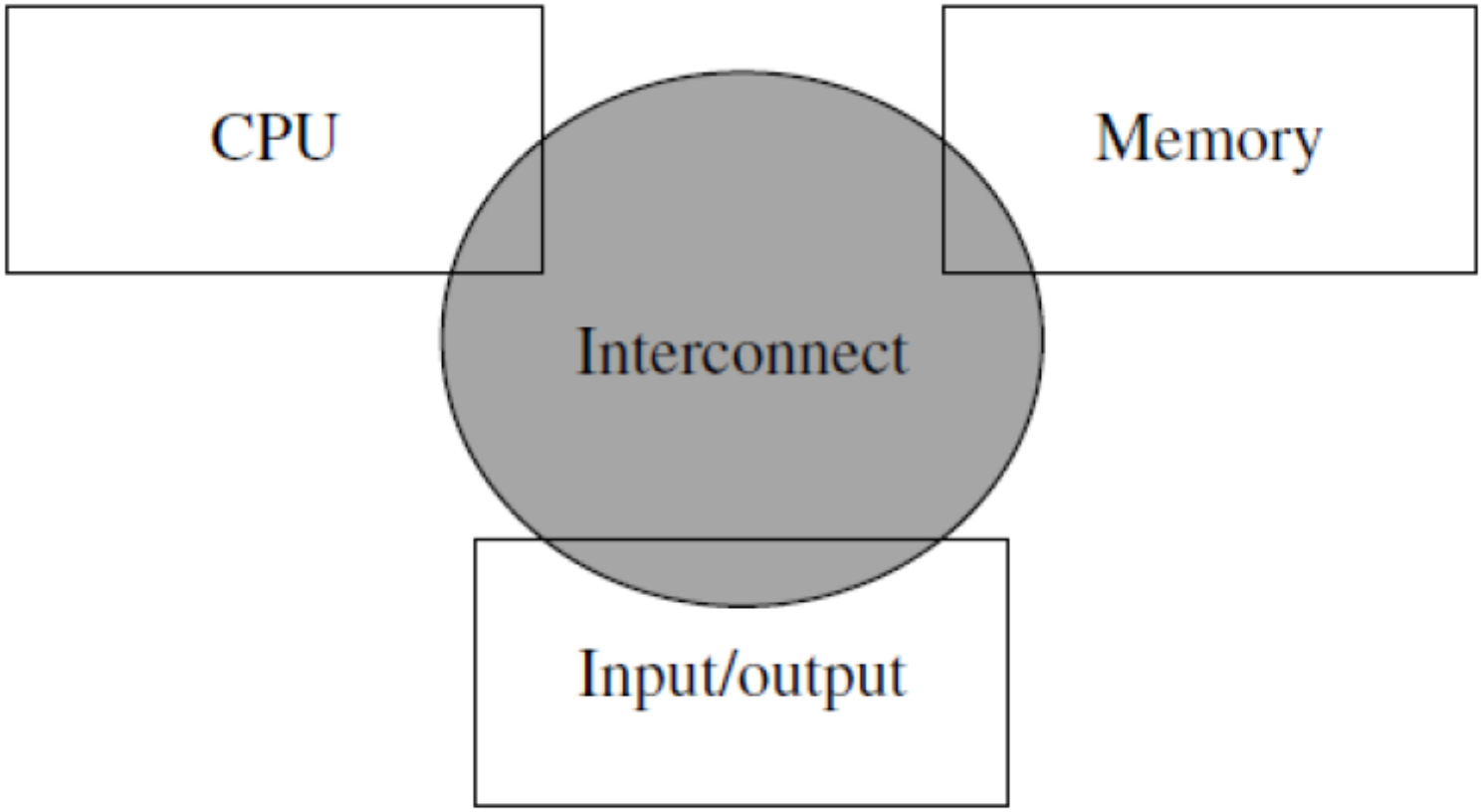
Programmers While Coding

It Doesn't Work..... Why?

It Work..... Why?

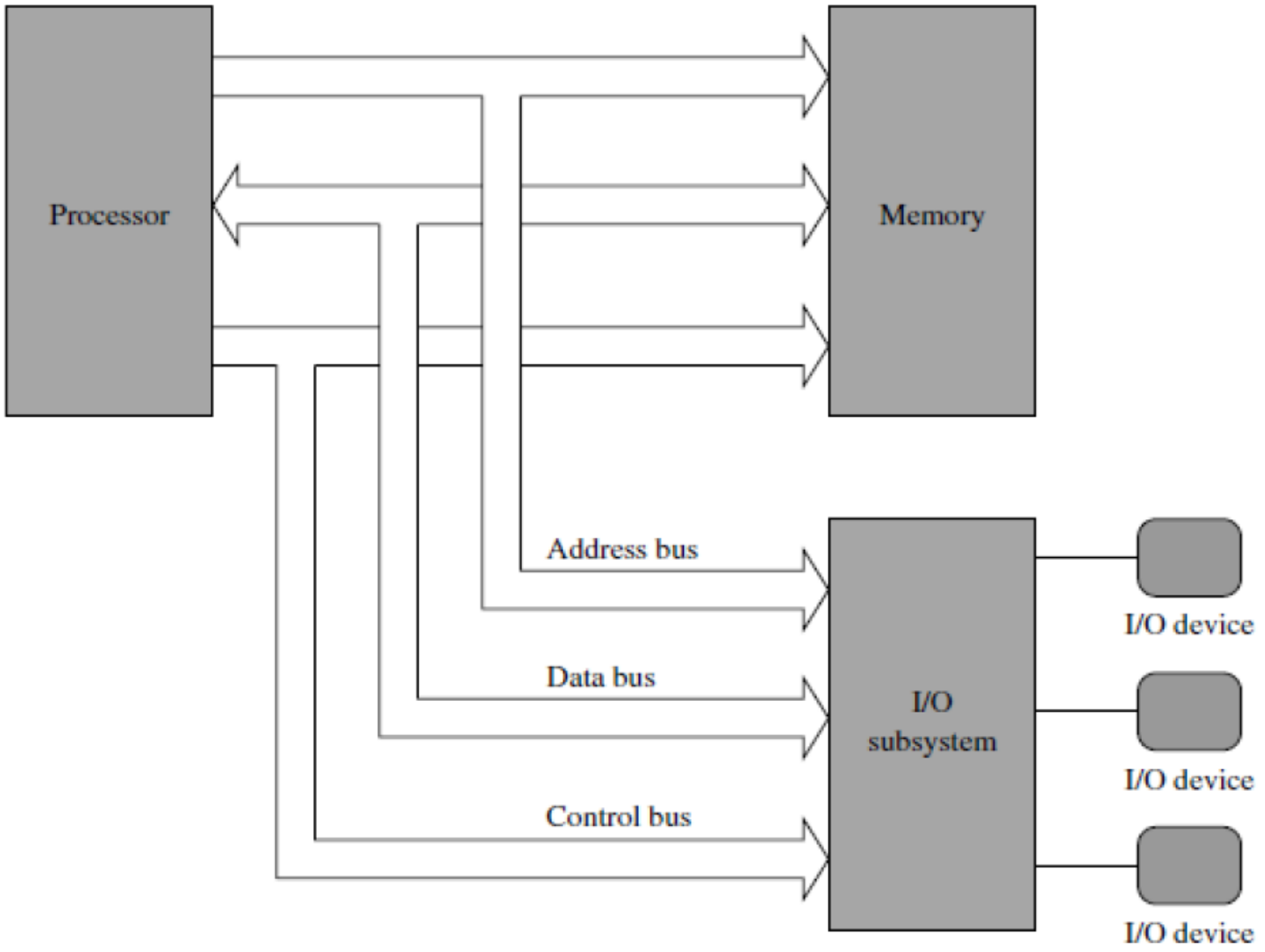
High level view of a Computer System



Basic Components of a Computer System

- A computer system has three main components:
  - central processing unit (CPU) or processor,
  - a memory unit,
  - input/output (I/O) devices
- These three components are interconnected by a system bus.
- The term “bus” is used to represent a group of electrical signals or the wires that carry these signal.

Simplified block diagram of a computer system



ARM Ltd

- Founded in November 1990
  - Spun out of Acorn Computers
- Designs the ARM range of RISC processor cores
- Licenses ARM core designs to semiconductor partners who fabricate and sell to their customers.
  - ARM does not fabricate silicon itself
- Also develop technologies to assist with the design-in of the ARM architecture
  - Software tools, boards, debug hardware, application software, bus architectures, peripherals etc

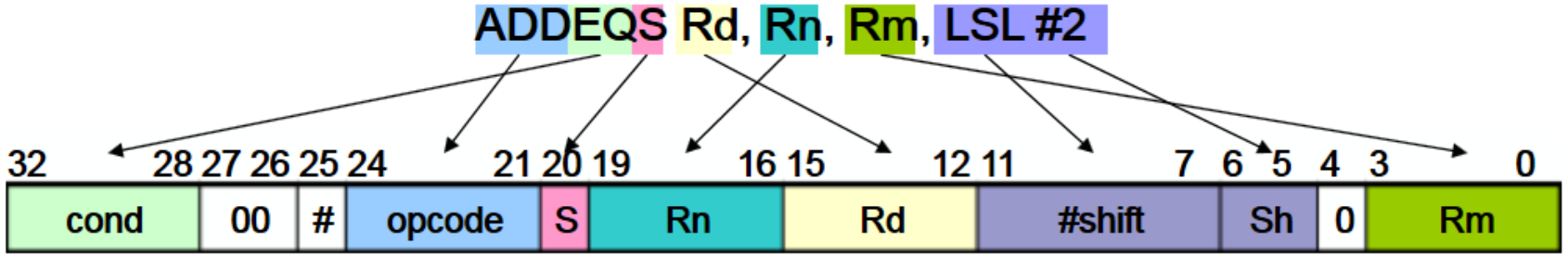
ARM

ARM Powered Products

RISC machine

- RISC architecture
  - Fixed instruction size (e.g., 32bit)
  - Load-store architecture
    - Operands must be located in registers
    - The operation result is put into register
  - Large register file
  - Simple addressing modes
- RISC organization
  - Hard-wired instruction decoding logic
  - Pipelined execution
  - Single-cycle execution

Instruction format



RISC machine

- Advantage
  - Simple hardware
    - Small die size
    - Low power consumption
  - Simple decoding
  - Higher performance
    - Easy to implement an effective pipelined structure
- Disadvantage
  - Poor code density
    - RISC has a fixed size of instruction format
    - Small number of instructions

About the ARM architecture

- The ARM architecture
  - RISC + additional features
  - Occupies almost 75% of 32bit embedded RISC microprocessor market
- Additional features of ARM
  - Auto-increment/decrement addressing modes
  - Single data-processing instruction can perform both ALU and shifter operations
  - Load/Store multiple instruction
  - Conditional execution

ARM architecture versions

Architecture	Family
ARMv1	ARM1
ARMv2	ARM2, ARM3
ARMv3	ARM6, ARM7
ARMv4	StrongARM, ARM7TDMI, ARM9TDMI
ARMv5	ARM7EJ, ARM9E, ARM10E, Xscale
ARMv6	ARM11, ARM Cortex-M
ARMv7	ARM Cortex-A, ARM Cortex-M, ARM Cortex-R
ARMv8	Support 64-bit addressing

"ARM Architecture." Wikipedia, The Free Encyclopedia.

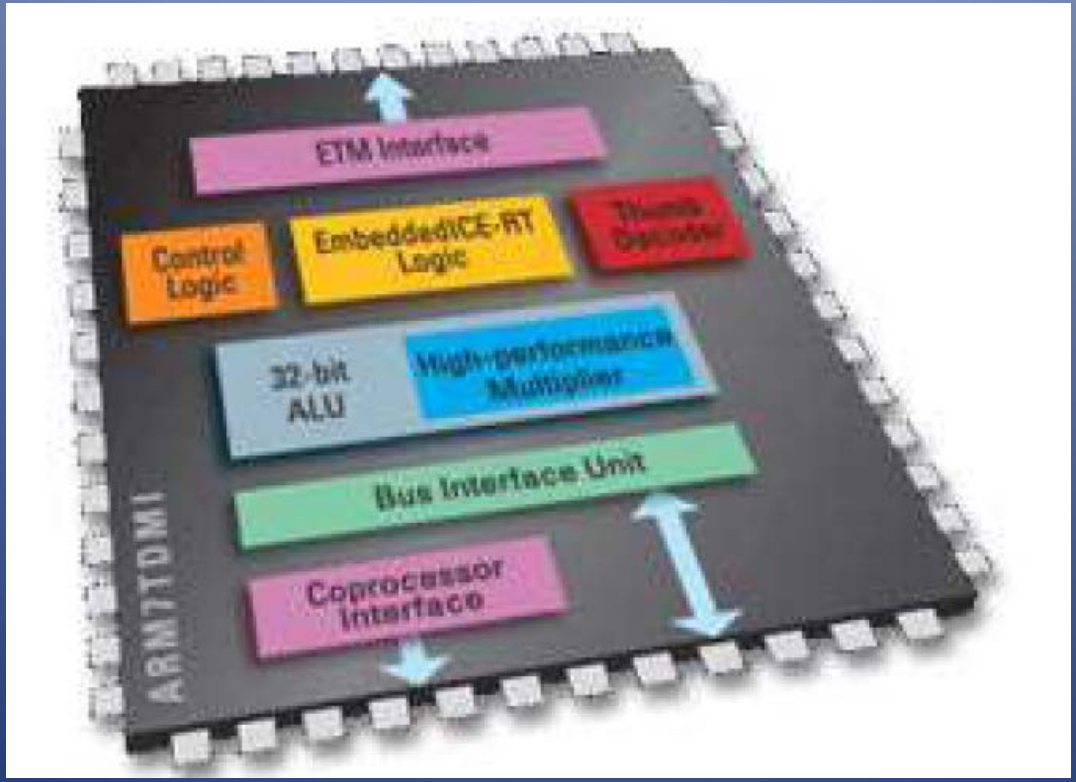
ARM7TDMI stands for

T: THUMB ;

D for on-chip Debug support, enabling the processor to halt in response to a debug request,

M: enhanced Multiplier, yield a full 64-bit result, high performance

I: Embedded ICE hardware (In Circuit emulator)



ARM7TDMI PROGRAMMER'S MODEL

Processor operating states

The ARM7TDMI processor has two operating states:

ARM 32-bit, word-aligned ARM instructions are executed in this state.

Thumb 16-bit, halfword-aligned Thumb instructions are executed in this state.

In Thumb state, the *Program Counter* (PC) uses bit 1 to select between alternate halfwords.

Note

Transition between ARM and Thumb states does not affect the processor mode or the register contents.

Thumb code is typically 65% of the size of ARM code, and provides 160% of the performance of ARM code when running from a 16-bit memory system. Thumb, therefore, makes the ARM7TDMI core ideally suited to embedded applications with restricted memory bandwidth, where code density is important.

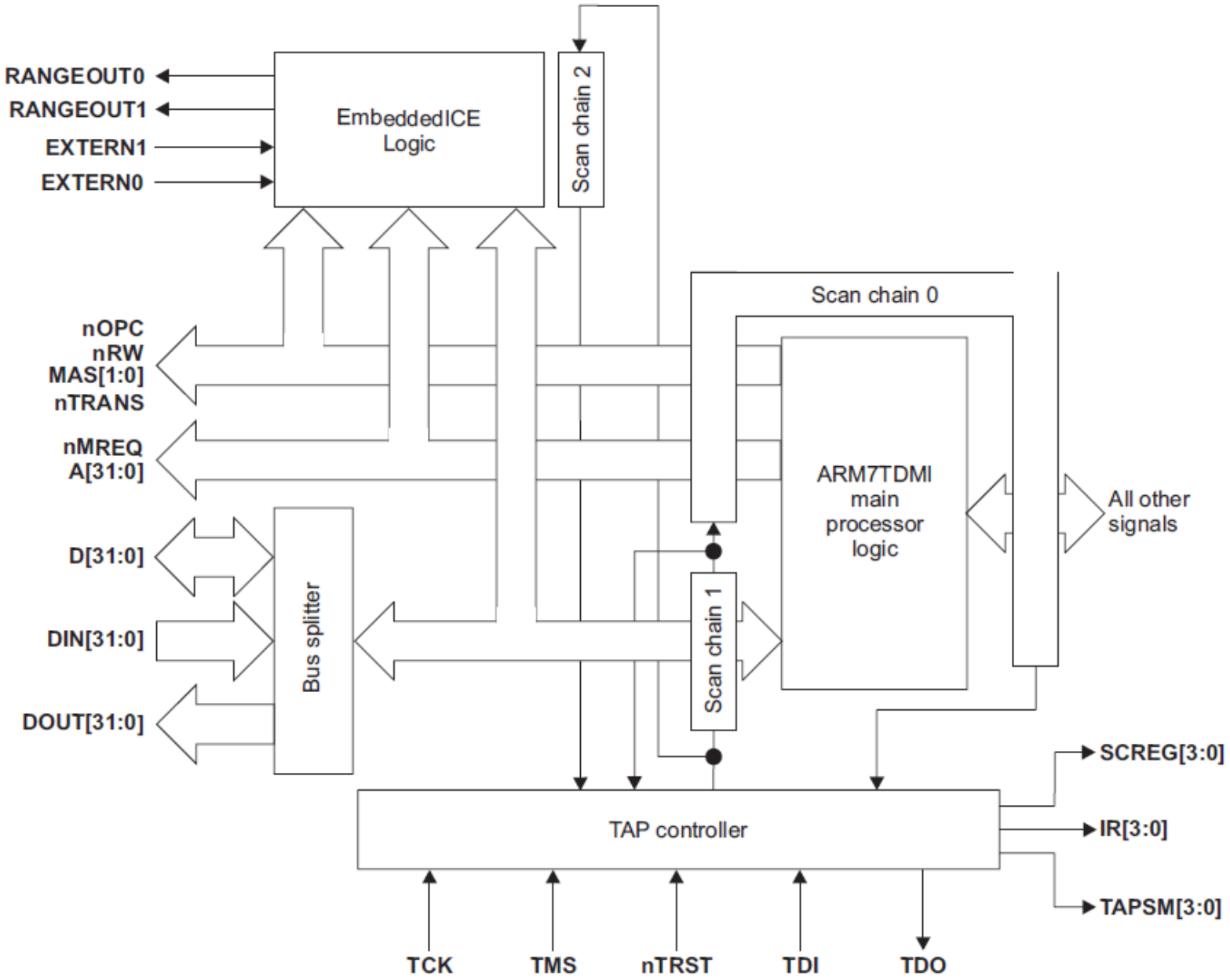
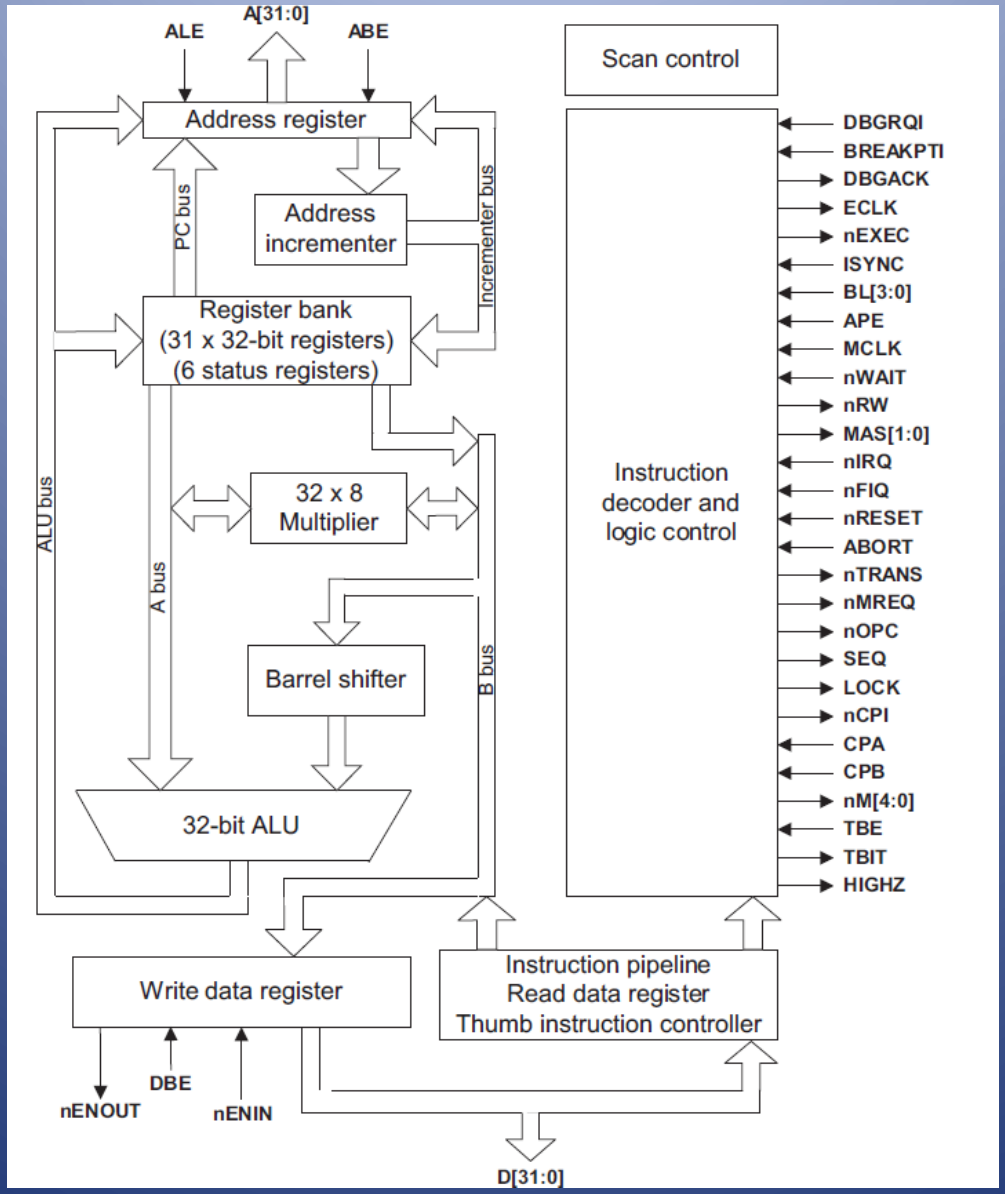


Figure ARM7TDMI processor block diagram

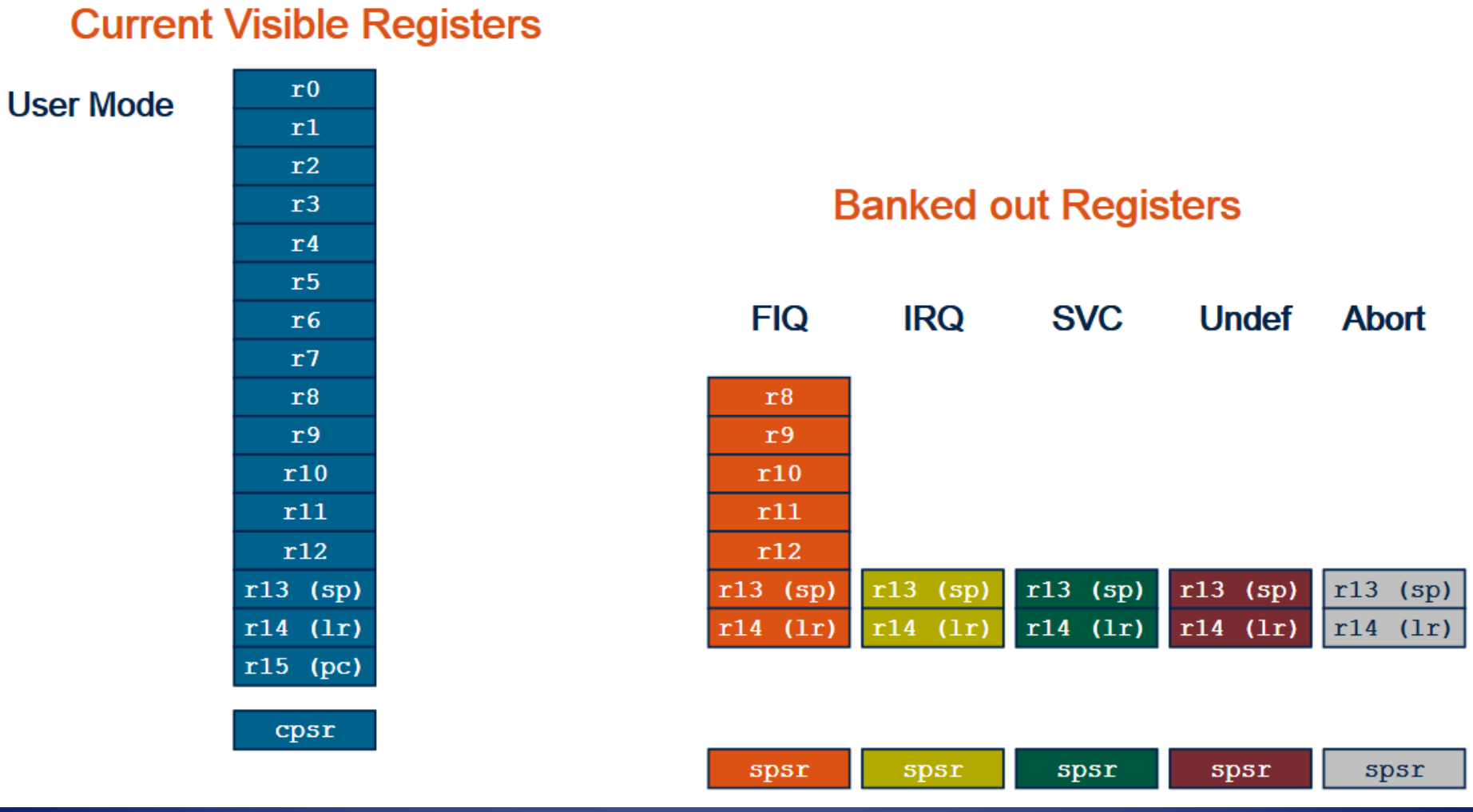
ARM7TDMI PROGRAMMER'S MODEL



REGISTERS

- **Visible registers**
  - 31 general-purpose registers, 6 program status registers
  - At any time, 16 general-purpose registers and one or two status registers are visible according to processor mode
- **General-purpose registers (GPR)**
  - Unbanked registers, R0-R7, R15
    - The same physical registers in all processor modes
  - Banked registers, R8-R14
    - The physical register referred to by each of them depends on the current processor mode
  - Special function of R13-15
    - Stack pointer (R13)
    - Link register (R14): save the return address
    - Program counter (R15): point to address of instruction to be fetched

REGISTERS


Current Visible Registers
User Mode
r0
r1
r2
r3
r4
r5
r6
r7
r8
r9
r10
r11
r12
r13 (sp)
r14 (lr)
r15 (pc)
cpsr
Banked out Registers
FIQ
IRQ
SVC
Undef
Abort
r8
r9
r10
r11
r12
r13 (sp)
r14 (lr)
r13 (sp)
r14 (lr)
r13 (sp)
r14 (lr)
r13 (sp)
r14 (lr)
r13 (sp)
r14 (lr)
spsr
spsr
spsr
spsr
spsr

ARM Processor Modes

- The ARM has seven basic operating modes:
  - User : unprivileged mode under which most tasks run
  - FIQ : entered when a high priority (fast) interrupt is raised
  - IRQ : entered when a low priority (normal) interrupt is raised
  - Supervisor : entered on reset and when a Software Interrupt instruction is executed
  - Abort : used to handle memory access violations
  - Undef : used to handle undefined instructions
  - System : privileged mode using the same registers as user mode

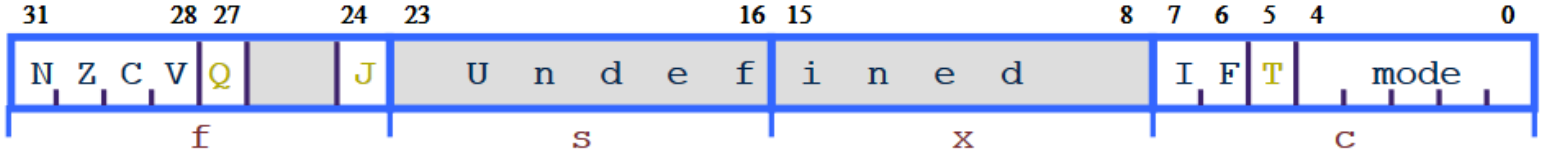
M[4:0]	Mode	Visible Thumb-state registers	Visible ARM-state registers
10000	User	r0–r7, SP, LR, PC, CPSR	r0–r14, PC, CPSR
10001	FIQ	r0–r7, SP_fiq, LR_fiq, PC, CPSR, SPSR_fiq	r0–r7, r8_fiq–r14_fiq, PC, CPSR, SPSR_fiq
10010	IRQ	r0–r7, SP_irq, LR_irq, PC, CPSR, SPSR_irq	r0–r12, r13_irq, r14_irq, PC, CPSR, SPSR_irq
10011	Supervisor	r0–r7, SP_svc, LR_svc, PC, CPSR, SPSR_svc	r0–r12, r13_svc, r14_svc, PC, CPSR, SPSR_svc
10111	Abort	r0–r7, SP_abt, LR_abt, PC, CPSR, SPSR_abt	r0–r12, r13_abt, r14_abt, PC, CPSR, SPSR_abt
11011	Undefined	r0–r7, SP_und, LR_und, PC, CPSR, SPSR_und	r0–r12, r13_und, r14_und, PC, CPSR, SPSR_und
11111	System	r0–r7, SP, LR, PC, CPSR	r0–r14, PC, CPSR

The program status registers

The ARM7TDMI processor contains a CPSR and five SPSRs for exception handlers to use. The program status registers:

- hold information about the most recently performed ALU operation
- control the enabling and disabling of interrupts
- set the processor operating mode.

Program Status Registers



- **Condition code flags**
  - N = Negative result from ALU
  - Z = Zero result from ALU
  - C = ALU operation Carried out
  - V = ALU operation oVerflowed
- **Sticky Overflow flag - Q flag**
  - Architecture 5TE/J only
  - Indicates if saturation has occurred
- **J bit**
  - Architecture 5TEJ only
  - J = 1: Processor in Jazelle state
- **Interrupt Disable bits.**
  - I = 1: Disables the IRQ.
  - F = 1: Disables the FIQ.
- **T Bit**
  - Architecture xT only
  - T = 0: Processor in ARM state
  - T = 1: Processor in Thumb state
- **Mode bits**
  - Specify the processor mode

3-Stage Pipeline ARM Organization

- **The 3-stage pipeline**
  - ARM processors up to the ARM7
  - Pipeline stages
    - 1. **Fetch**: The instruction is fetched from memory and placed in the instruction pipeline.
    - 2. **Decode**: The instruction is decoded and the datapath control signals prepared for the next cycle. In this stage, the instruction 'owns' the decode logic but not the datapath.
    - 3. **Execute**: The instruction 'owns' the datapath; the register bank is read, an operand shifted, the ALU result generated and written back into a destination register.
  - At any one time, three different instructions may occupy each of these stages: The hardware in each stage has to be capable of independent operation.