

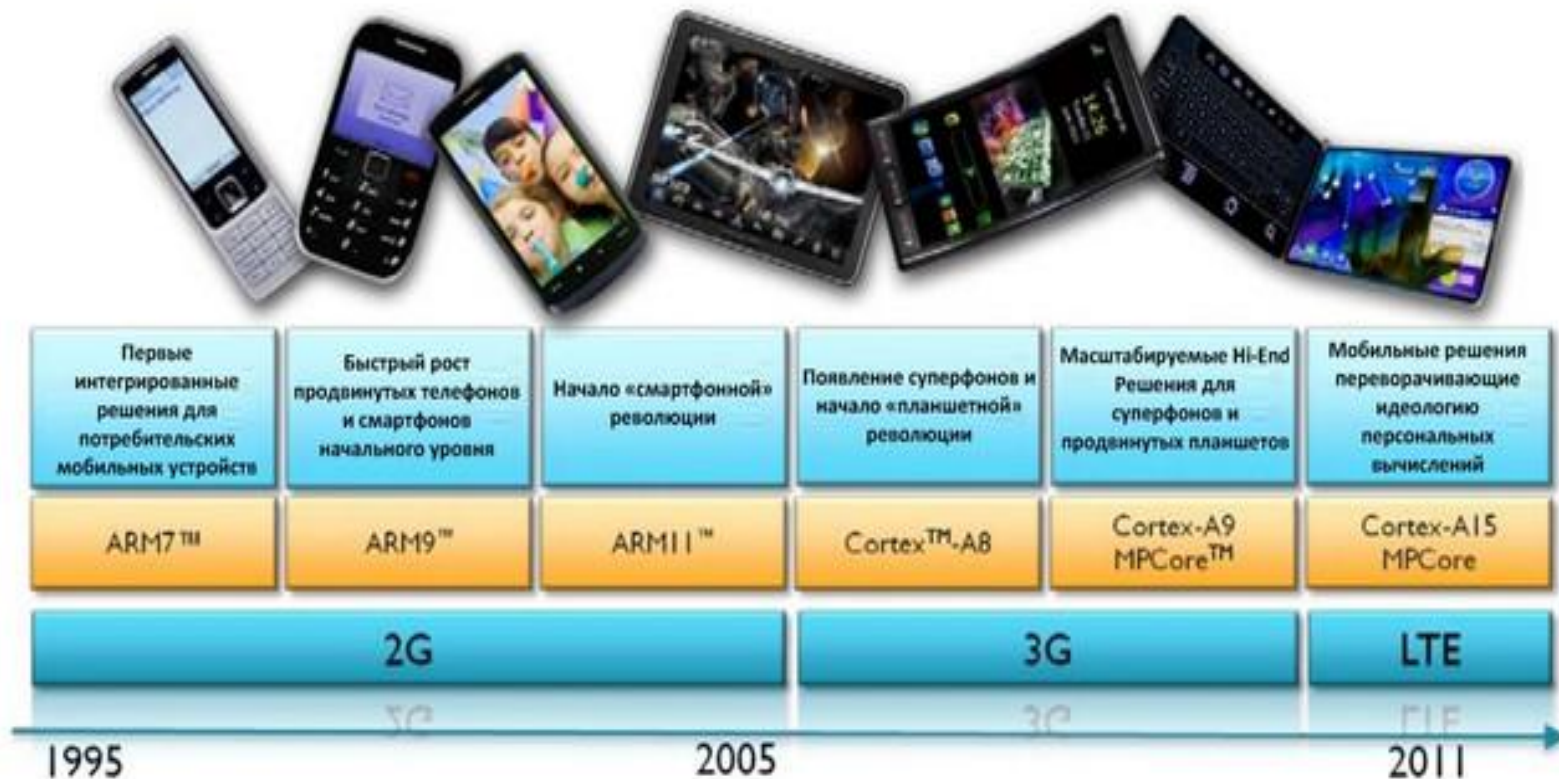
ARM-процесори

Вступ

CISC (Complex instruction set computer) - комп'ютер із комплексним набором команд

RISC (Restricted (reduced) instruction set computer) - комп'ютер зі скороченим набором команд

ARM - Acorn RISC Machine



ARM1

Перший процесор ARM1 для Acorn зробила компанія-партнер VLSI

Технологія ASSP (Application-specific standard products) передбачає розробку простих, але в той же час універсальних по застосуванню компонентів - наприклад, апаратних декодерів звуку і відео.

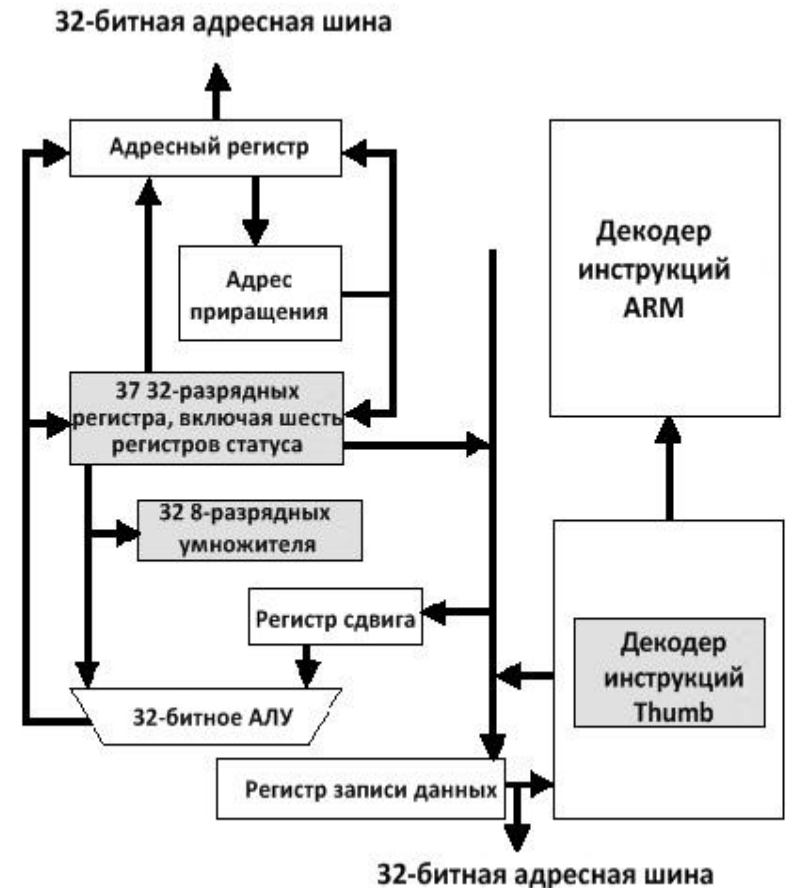
Технологія ASIC (Application-specific integrated circuit) - в протилежність ASSP, та й процесорним системам загального призначення - передбачає створення інтегральних мікросхем, що спеціалізуються на вирішенні деякого обмеженого кола завдань. До ASIC-рішень можна віднести роутери, мобільні телефони й ігрові консолі.



Ядро ARM7

У 1994 році ARM випустила процесорне ядро ARM7.

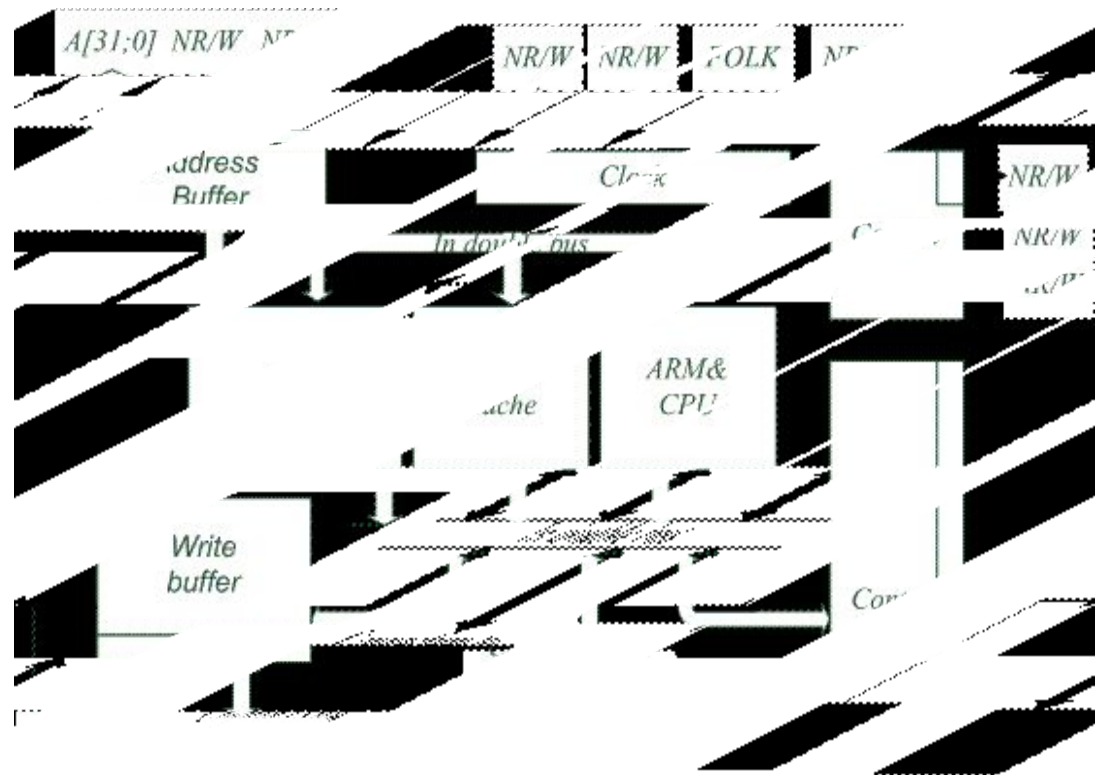
Особливістю структурної схеми ядра **ARM7** є наявність декодеру інструкцій **Thumb** та декодеру інструкцій **ARM**. За допомогою їх використання вдається компресувати 32-розрядні команди у 16-розрядні, а потім безпосередньо перед їх виконанням декомпресувати.



Блок-схема ядра ARM7TDMI

Ядро ARM7 містить

- **ARM7 CPU** - 32-розрядний RISC процесор (32-розрядні шини даних і адреси);
- **Cache** – спільна кеш-пам'ять команд і даних ємністю 4 кБайта;
- **MMU** – пристрій керування пам'яттю (для організації віртуальної пам'яті, для розбивання пам'яті на сторінки);
- **Address Buffer** – буфер адреса ;
- **Write Buffer** – буфер запису;
- **Clock** – тактовий генератор (25 МГц);
- **Control** – схема керування і синхронізації ;
- **Control Coproc** – схема керування під'єднанням співпроцесорів.



Система команд ядра ARM7

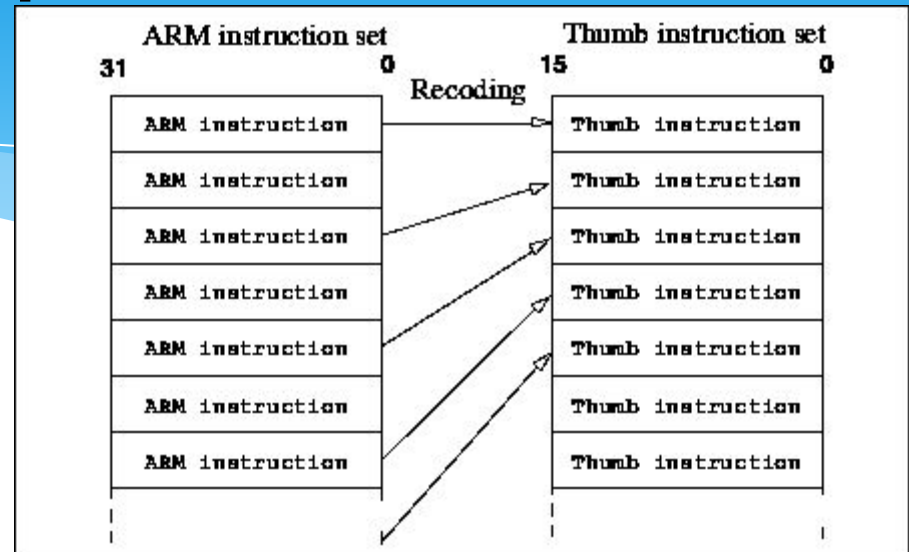
32-розрядна система команд ядра **ARM7** містить одинадцять базових типів команд:

- Два типи використовують вбудований арифметико-логічний пристрій, циклічний пристрій зсуву і перемножувач при операціях над даними в банку з 31 регістра, форматом по 32 розряду кожен;
- Три класи команд управління переміщенням даних між пам'яттю і регістрами, один оптимізований на забезпечення гнучкості адресації, інший під швидке контекстне перемикання і третій під підкачку даних;
- Три типи команд управляють потоком і рівнем привілею виконання;
- Три типи призначені для управління зовнішніми співпроцесорами, що дозволяє розширити функціональні можливості системи команд за межами ядра.

Система команд ARM добре обробляється компіляторами мов високого рівня. На відміну від деяких RISC процесорів, процесор ARM7, при виникненні необхідності в деякому зменшенні обсягу кодів, допускає програмування і на асемблері.

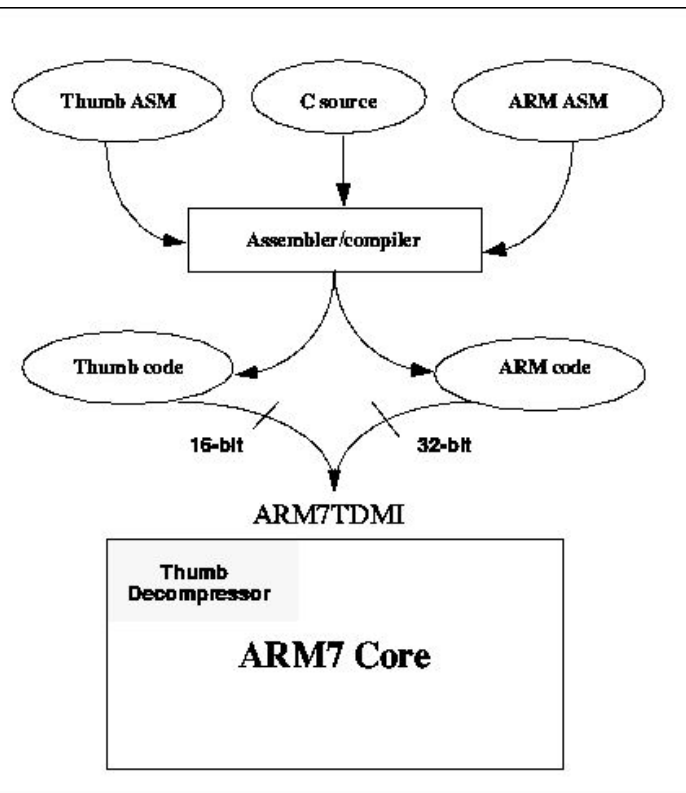
Концепція Thumb

Система команд **Thumb** містить 36 команд, похідних від стандартної 32-розрядної системи команд **ARM**, перекодованих в 16-розрядні коди.



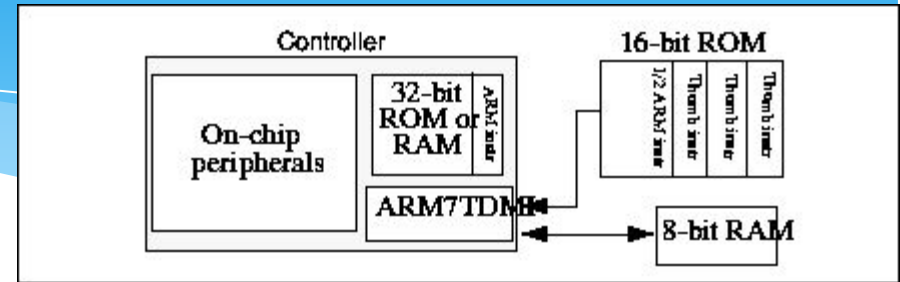
У процесі виконання ці нові 16-розрядні **Thumb** коди декомпресуються процесором у відповідні еквівалентні команди **ARM**, які потім і виконуються ядром **ARM**

Комплект дозволяє програмісту писати і розміщувати в пам'яті системи коди **ARM**, коди **Thumb** або усі разом

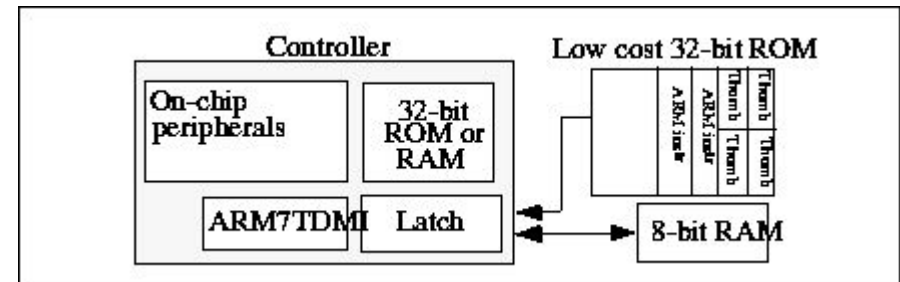


Приклади конфігурації системи

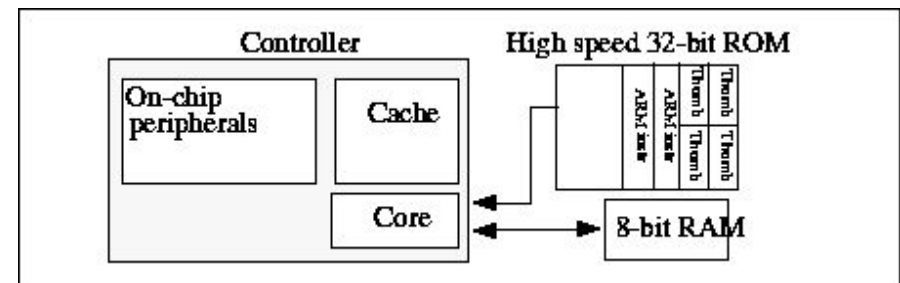
Варіант конфігурації мікроконтролера з ядром **ARM7TDMI** та 16-розрядною системою пам'яті



Варіант конфігурації мікроконтролера з ядром **ARM7TDMI** та 32-розрядною ROM пам'яттю



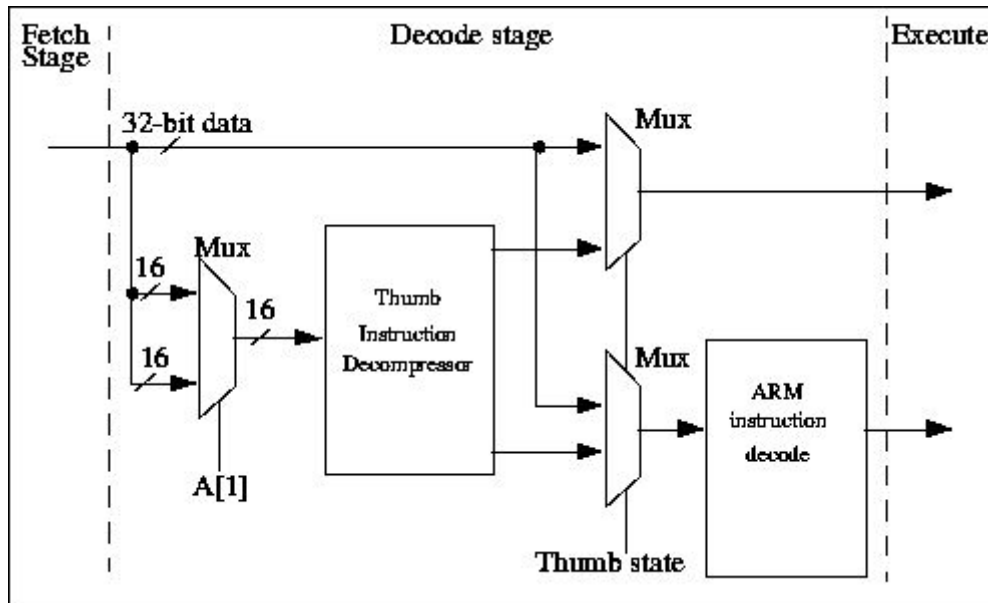
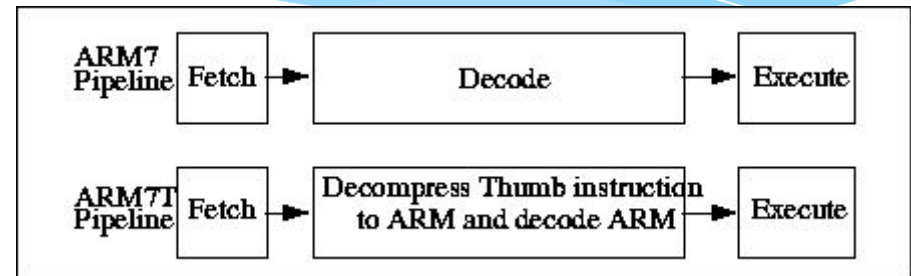
Варіант конфігурації мікроконтролера з ядром **ARM7TDMI** та високопродуктивною 32-розрядною системою



Особливості роботи першого Thumb-орієнтованого ядра ARM7TDMI

Основний додаток до архітектури **ARM**, що забезпечує підтримку системи команд **Thumb** - декомпресор **Thumb**. Першим ядром **ARM**, яке містило декомпресор, стало ядро **ARM7TDMI**.

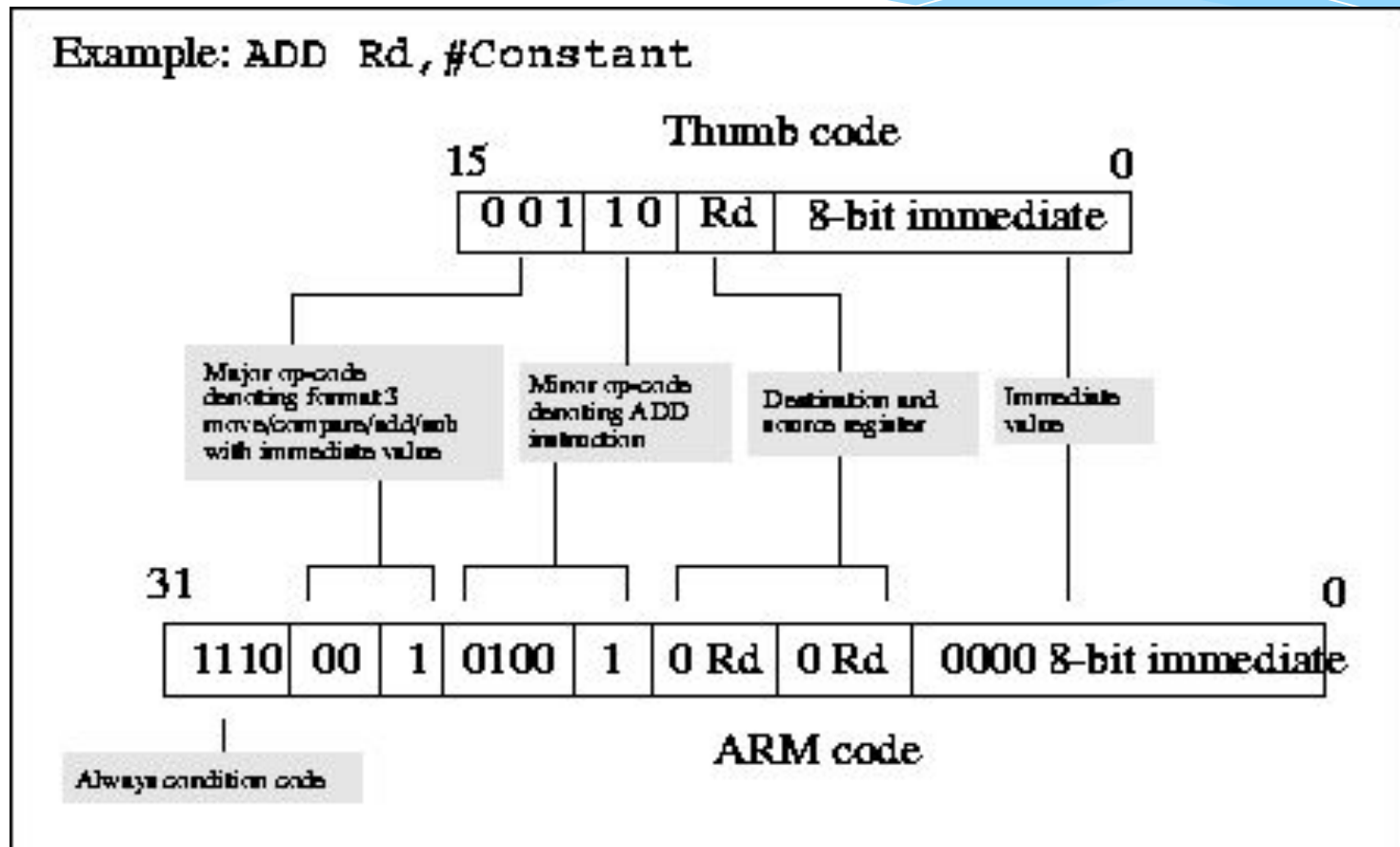
3-рівневий конвеєр з фазами
вибірки, декодування і виконання



Процес **Thumb**-
декодування і декомпресії

Приклад Thumb-коду

У **Thumb**-стані мультиплексори направляють Thumb-команди через логіку декомпресії **Thumb**, розгортаючи команду **Thumb** в її еквівалент **ARM** команди. Потім команда **ARM** виконується в нормальному режимі.



Переривання ядра ARM7TDMI

1. Reset
2. Data abort
3. FIQ
4. IRQ
5. Prefetch abort
6. Undefined instruction
7. Software interrupt

Стани ARM і THUMB

У стані **ARM** процесор виконує 32-розрядні команди, у стані **THUMB** — 16-розрядні команди.

У стані **ARM** процесор може функціонувати в одному з наступних режимів:

1. User
2. Supervisor
3. System
4. IRQ
5. FIQ (Fast IRQ)
6. Abort
7. Undefined

Регістри загального призначення та лічильник програми в стані ARM

Системний режим і режим користувача	Режим швидкого переривання	Супервізорний режим	Аварійний режим	Режим переривання	Невизначений режим
r0	r0	r0	r0	r0	r0
r1	r1	r1	r1	r1	r1
r2	r2	r2	r2	r2	r2
r3	r3	r3	r3	r3	r3
r4	r4	r4	r4	r4	r4
r5	r5	r5	r5	r5	r5
r6	r6	r6	r6	r6	r6
r7	r7	r7	r7	r7	r7
r8	r8_fiq	r8	r8	r8	r8
r9	r9_fiq	r9	r9	r9	r9
r10	r10_fiq	r10	r10	r10	r10
r11	r11_fiq	r11	r11	r11	r11
r12	r12_fiq	r12	r12	r12	r12
r13	r13_fiq	r13_svc	r13_abt	r13_irq	r13_und
r14	r14_fiq	r14_svc	r14_abt	r14_irq	r14_und
r15 (PC)	r15 (PC)	r15 (PC)	r15 (PC)	r15 (PC)	r15 (PC)

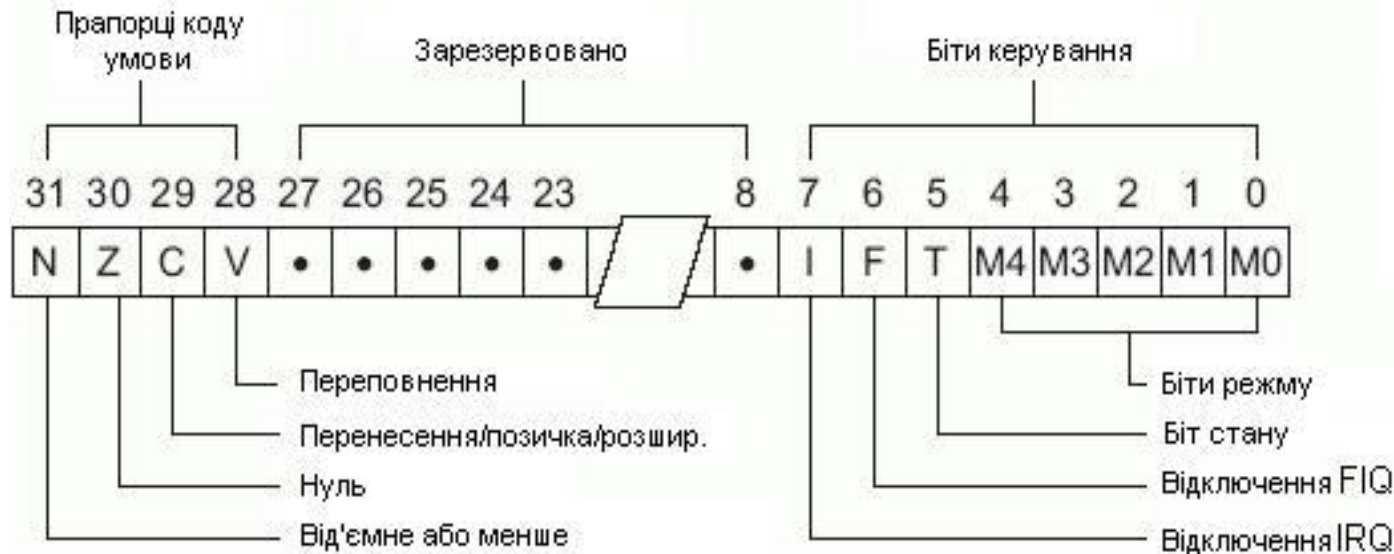
Регістри статусу програми в стані ARM

CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR_fiq	SPSR_svc	SPSR_abt	SPSR_irq	SPSR_und

 - банкований регістр

Регістр стану CPSR

- М4-0 - визначають режим роботи процесора відповідно до таблиці;
- Т - задає стан процесора: ARM (при $T = 0$) або THUMB (при $T = 1$);
- І, F - маскують (забороняють обробку) переривань IRQ і FIQ відповідно
- N, Z, C, V - є ознаками знака (N), нуля (Z), перенесення (C) і переповнення (V), значення яких установлюються відповідно до результату чергової операції. У таблиці М4-М0 зображено відповідність конфігурації бітів М4-0 режимам процесора.



M4-0					Режим
1	0	0	0	0	User
1	0	0	0	1	FIQ
1	0	0	1	0	IRQ
1	0	0	1	1	Supervisor
1	0	1	1	1	Abort
1	1	0	1	1	Undefined
1	1	1	1	1	System

Стан Thumb

Набір реєстрів у стані **Thumb** скорочений - він є підмножиною реєстрового банку в стані **ARM**. Програміст має доступ до:

- 8 реєстрів загального призначення r0-r7
- Лічильника програм PC
- Показчика стека SP
- Регістру зв'язку LR
- Регістру поточного стану програми CPSR.

Регістри загального призначення та лічильник програми в стані Thumb

Системний режим режим користувача	Режим швидкого переривання	Супервізорний режим	Аварійний режим	Режим переривання	Невизначений режим
r0	r0	r0	r0	r0	r0
r1	r1	r1	r1	r1	r1
r2	r2	r2	r2	r2	r2
r3	r3	r3	r3	r3	r3
r4	r4	r4	r4	r4	r4
r5	r5	r5	r5	r5	r5
r6	r6	r6	r6	r6	r6
r7	r7	r7	r7	r7	r7
SP	SP_fiq	SP_svc	SP_abt	SP_irq	SP_und
LR	LR_fiq	LR_svc	LR_abt	LR_irq	LR_und
PC	PC	PC	PC	PC	PC

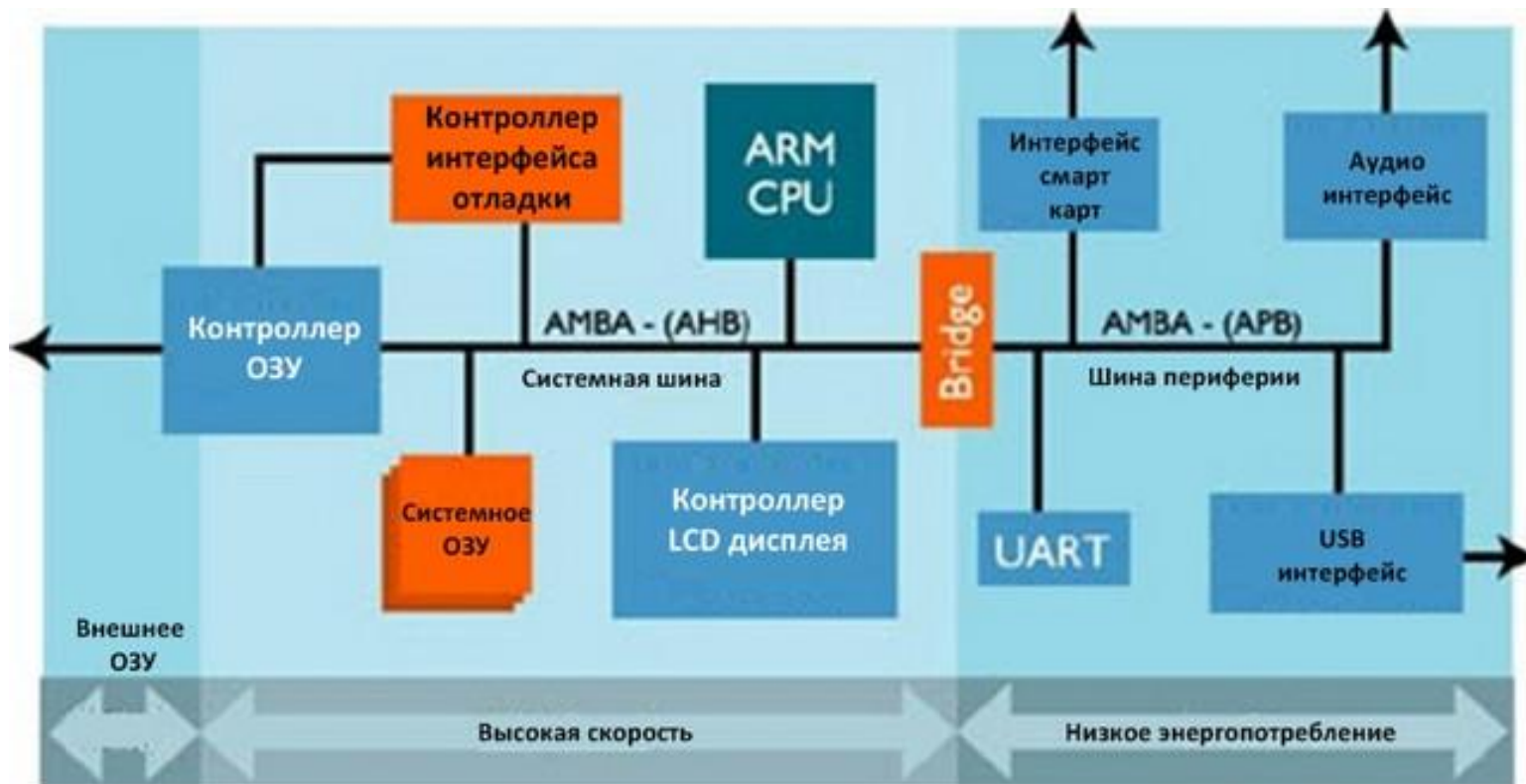
Регістри статусу програми в стані Thumb

CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR_fiq	SPSR_svc	SPSR_abt	SPSR_irq	SPSR_und

 - банкований реєстр

Шина AMBA

Всю цю периферію на чіпі з ARM-ядром об'єднує шина **AMBA (Advanced Microcontroller Bus Architecture)**. Шина **AMBA**, в свою чергу, ділиться на системну шину (**AHB**) і шину периферійних пристроїв (**APB**). Завданням AHB є забезпечення високошвидкісного обміну між ядром ARM і контролерами пам'яті і LCD-екрану, в той час як мета APB - зниження енергоспоживання при роботі з підключеною периферією.



Серія Cortex

В основу процесорної лінійки **Cortex** компанія ARM поклала як нову архітектуру **ARMv9**

Thumb-2 - наступне покоління успішної системи ущільнення коду Thumb. **Thumb-2** розширила набір 16-розрядних Thumb-команд і доповнила його повноцінними 32-розрядними інструкціями.

The ARM logo consists of the word "ARM" in white, bold, sans-serif capital letters, centered within a solid blue square. A small registered trademark symbol (®) is located at the top right of the letter "M".

ARM®

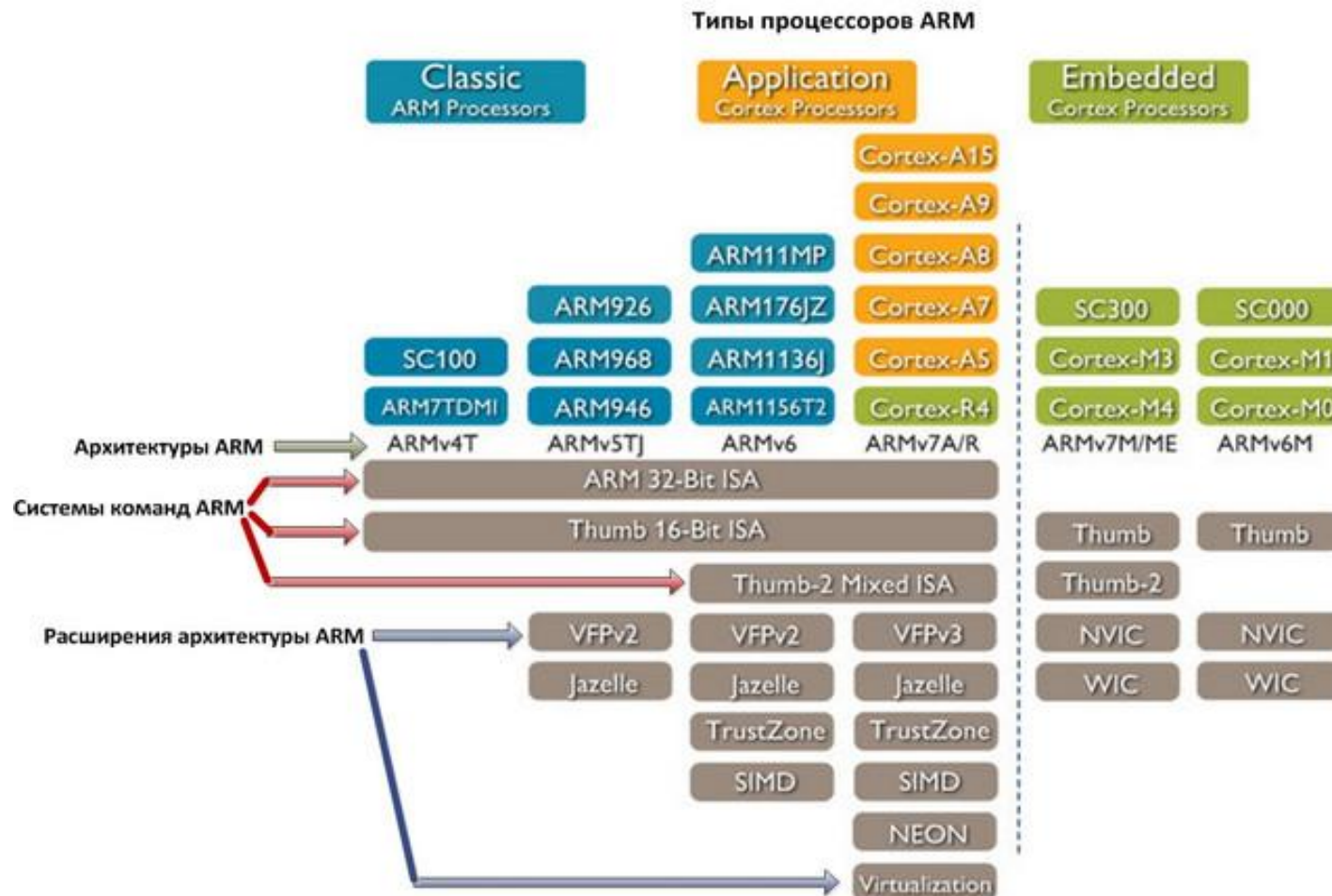
The NEON logo features the word "NEON" in blue, bold, sans-serif capital letters, followed by a trademark symbol (™). The text is positioned on a white rectangular background that is slightly offset to the right and bottom relative to the ARM logo.

NEON™

NEON є співпроцесором ARMv9. Володіючи незалежним конвеєрним модулем, власними тридцятьма двома 64-розрядними і шістнадцятьма 128-розрядними регістрами даних, **NEON** з легкістю працює з такими завданнями, як кодування і декодування відео та звуку і обробка 2D-і 3D-графіки. При цьому модуль **NEON** тісно інтегрується з ARM-ядром, а це значить, що в багатоядерної архітектурі MPCore число модулів **NEON** збігається з числом процесорних ядер. Унікальна архітектура **NEON** забезпечує його продуктивність, як мінімум в три рази перевищує продуктивність класичних ARM-ядер на архітектурі **ARMv7**

Типи процесорів

Кожна наступна лінійка процесорів ARM підтримує технологічні рішення попередників і включає в себе нові технології.



Cortex-A, Cortex-R, Cortex-M

Cortex-A (від application) - сімейство процесорів, орієнтованих на ринок споживчої електроніки і здатних вирішувати широкий спектр завдань, яким сучасні користувачі так люблять навантажувати свої гаджети.

Cortex-R (від real time) - серія мікропроцесорів, оптимізованих для виконання обчислень в режимі реального часу.

Cortex-M (від eMbedded) - лінійка Cortex-процесорів, які прийшли на зміну 8 - і 16-розрядних мікроконтролерів вбудованих систем.

