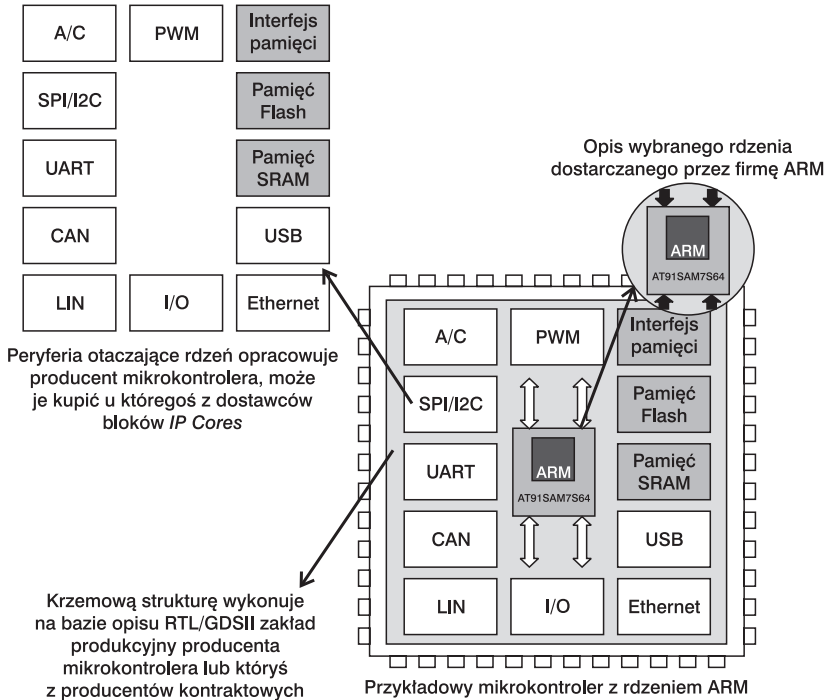


1. Rdzenie ARM

Historia rdzeni ARM sięga 1985 roku, gdy firma Acorn Computer Group opracowała pierwszy na świecie komercyjny procesor RISC, który po dość istotnych modyfikacjach trafił w roku 1987 do domowych komputerów PC produkowanych przez ACG. Komputery Acorn RISC PC miały własny, „okienkowy” system operacyjny RISC OS, którego możliwości były co najmniej porównywalne z ówczesnym systemem Windows. Pomimo dobrych parametrów oferowanych komputerów, pozostały one praktycznie nieznanne poza granicami Wielkiej Brytanii. Zmusiło to firmę do poszukiwania alternatywnych dróg rozwoju. W roku 1990 powstała firma córka o nazwie Advanced RISC Machines (ARM), której zadaniem było promowanie i rozwój udanych procesorów RISC. Do spółki przystąpiła firma VLSI, której ogromnym atutem była... fabryka struktur półprzewodnikowych. Dość szybko (w roku 1991, po wprowadzeniu do sprzedaży procesorów z rodziny ARM6) okazało się, że lepszym interesem od produkcji krzemu jest sprzedaż licencji na używanie zoptymalizowanych i dobrze oprogramowanych rdzeni procesorów RISC. Pojęcie *Intellectual Property Cores* nie funkcjonowało w świadomości elektroników tak mocno jak teraz, ale działalność ARM stopniowo zawężała się do prac badawczo rozwojowych. Firma zrezygnowała z produkcji zarówno krzemowych struktur, jak i projektowania całych układów scalonych. Obecnie firma ARM Holdings plc zajmuje się tylko i wyłącznie tworzeniem opisów struktur rdzeni ARM w językach HDL (*Hardware Description Language* – język opisu sprzętu), czyli tzw. *IP Cores* (*Intellectual Property Cores*). To właśnie ich sprzedaż jest źródłem zysków, jakie firma czerpie ze swojej działalności. Same układy scalone, które można znaleźć w ofertach wielu producentów (mikrokontrolery, mikroprocesory, systemy SoC) są produkowane przez firmy, które zakupiły u ARM Holdings plc licencje na wykorzystywanie zaprojektowanych przez tę firmę rdzeni. Tworząc mikrokontroler lub układ SoC, inżynierowie pracujący w tych firmach integrują rdzeń z odpowiednimi blokami peryferyjnymi, pamięciami RAM i Flash (w typowych konfiguracjach) i innymi elementami, jakie zamierzają umieścić w strukturze układu. Następnie stworzone przez nich opisy RTL (*Register Transfer Language* – rodzaj języka HDL pozwalający precyzyjnie opisać transfery pomiędzy rejestrami w cyfrowym układzie scalonym, takim jak mikrokontroler lub mikroprocesor, dokonywane przez poszczególne instrukcje), na podstawie których wytwórca układów scalonych tworzy konkretną strukturę krzemową mikrokontrolera, mikroprocesora lub układu SoC. Proces ten przedstawiono schematycznie na **rysunku 1.1**. Pierwsza znacząca w rozwoju firmy sprzedaż licencji na rdzenie ARM miała miejsce w roku 1992 – prawa do implementowania w swoich układach rdzeni ARM6 zakupiły firmy GEX Plessey i Sharp. W 1993 roku do grona firm korzystających z rdzeni ARM dołączyły firmy Cirrus Logic i Texas Instruments, a w kolejnych latach wiele kolejnych firm. ARM nieustannie rozwija swoją ofertę – rodzina ARM7 jest obecna na rynku od roku 1993, a w roku 1997 wprowadzono rdzeń ARM9. Dzięki temu są dostępne różne warianty rdzeni przystosowanych do pracy w różnych, także specyficznych aplikacjach. Wśród firm, które nabyły prawo do wykorzystywania we własnych opraco-



Rys. 1.1. Proces tworzenia mikrokontrolera z rdzeniem ARM

waniach rdzeni ARM można wymienić m.in.: Agilent, Asahi Kasei Microsystems, Alcatel, Altera, Atmel, Broadcom, Chip Express, Cirrus Logic, Digi, eSilicon, Freescale, Fujitsu, GEC Plessey, Global UniChip, HP, Hyundai, IBM, Intel, ITRI, LG Semicon, LSI Logic, Lucent, Luminary Micro, Matsushita, Micrel, Micronas, Mitsubishi, Motorola, NEC, OKI, Philips, Qualcomm, Rockwell, Rohm, Samsung, Sanyo, Seagate, Seiko Epson, Sharp, Sony, STMicroelectronics, Symbios Logic, Texas Instruments, Yamaha, Zeevo, ZTEIC, Zylogic. Rdzenie opracowywane przez ARM Holdings plc były pierwotnie przeznaczone wyłącznie do stosowania w układach ASIC (*Application Specific Integrated Circuit* – układ scalony opracowany specjalnie na potrzeby określonego zastosowania) produkowanych na masową skalę. Dzięki relatywnie prostej i nowoczesnej architekturze, specjalistycznemu wyposażeniu rdzenia i optymalizacji pod kątem możliwości technologii produkcyjnych dystrybuowanych rdzeni IP producenci układów scalonych mogli relatywnie niewielkim kosztem wbudować w nie 32-bitowe procesory RISC, których szybkość działania w typowych dla układów ASIC procesach technologicznych była znacznie większa niż w rozwiązaniach opartych na klasycznych procesorach.

Kolejnym poważnym argumentem przemawiającym na korzyść rdzeni opracowanych przez firmę ARM Holdings plc był niewielki pobór mocy. W zależności od technologii zastosowanej podczas produkcji, współcześnie produkowane CPU z rodziny ARM7 pobierają 0,025...0,15 W/MHz, co przy przeciętnej prędkości działania wynoszącej 0,8...1 MIPS/MHz umożliwia budowanie wydajnych systemów obliczeniowych zasilanych bateryjnie.

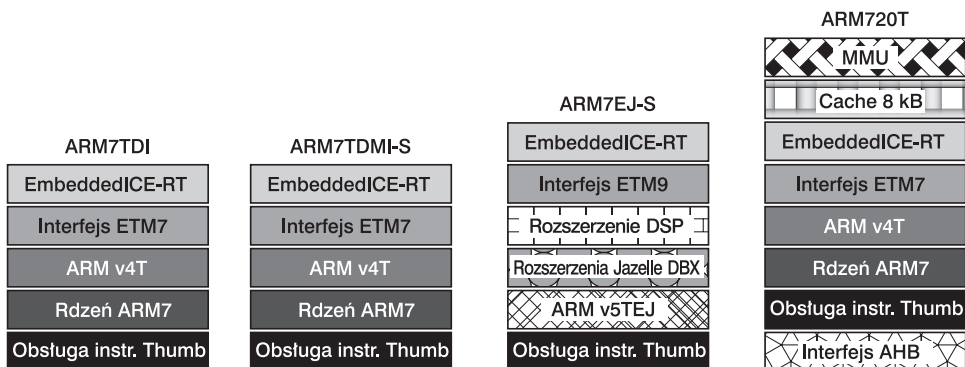
Pierwotny zamiar firmy ARM Holdings plc – czyli opracowanie procesora dla urządzeń i aplikacji wymagających dużej wydajności i do tego mobilnych – spowodował, że przez wiele lat o ARM-ach dużo się słyszało, dość często spotykało (choćby w telefonach komórkowych, grach multimedialnych czy palmtopach), ale mało kto miał możliwość zastosowania ich w praktyce. Wynikało to między innymi z tego, że producenci półprzewodników długo nie byli zainteresowani wprowadzaniem do produkcji nowej rodziny mikrokontrolerów, praktycznie bowiem każda licząca się na rynku firma oferowała własne rodziny mikrokontrolerów, więc tworzenie silnej konkurencji własnym wyrobom nie było uzasadnione.

Pierwszego wyłomu w takim postępowaniu dokonał Atmel w roku 1998, wprowadzając do produkcji prymitywne – z dzisiejszego punktu widzenia – procesory AT91M, w których producent wykorzystał rdzeń ARM7TDMI, ale brak wewnętrznej pamięci Flash uniemożliwił ich szybką popularyzację.

Kolejnym, tym razem przełomowym, krokiem w kierunku wprowadzenia „ARM-ów” pod strzechy był początek roku 2004, kiedy Philips wprowadził do sprzedaży mikrokontrolery z rodziny LPC2000. Układy te były wyposażone w wewnętrzną pamięć Flash, którą można było programować bez konieczności stosowania jakichkolwiek zewnętrznych narzędzi. Ponadto w mikrokontrolerach LPC2000 zintegrowano wiele typowych dla małych mikrokontrolerów bloków funkcjonalnych (m.in. interfejsy SPI, I²C, UART, przetworniki A/C). Zazwyczaj niemrawa na rynku popularnym firma Philips zaskoczyła innych producentów – do ataku w tym samym kierunku ruszyli z dużym opóźnieniem. Obecnie konstruktorzy chcący korzystać z mikrokontrolerów z rdzeniem ARM mają komfortową sytuację, w sprzedaży są bowiem dostępne różne wersje mikrokontrolerów z rdzeniami Cortex-M3, ARM7 oraz ARM9.

W nazewnictwie przyjętym przez firmę ARM panuje twórcze roztargnienie, które może wprawić w poważne zakłopotanie konstruktorów podczas ich pierwszych kontaktów z nowymi mikrokontrolerami. Czym się objawia to „roztargnienie”?

Po pierwsze, trzeba sobie uzmysłowić, że hasło mikrokontroler z rdzeniem „ARM7” lub „ARM9” niesie bardzo ogólną informację o tym układzie, na podstawie której nie można powiedzieć, czy jest on odpowiedni do wymagań stawianych przez aplikację. Wynika to z faktu, że w ramach rodzin rdzeni takich jak ARM Cortex, ARM7, ARM9 czy ARM10 występują różne architektury, których



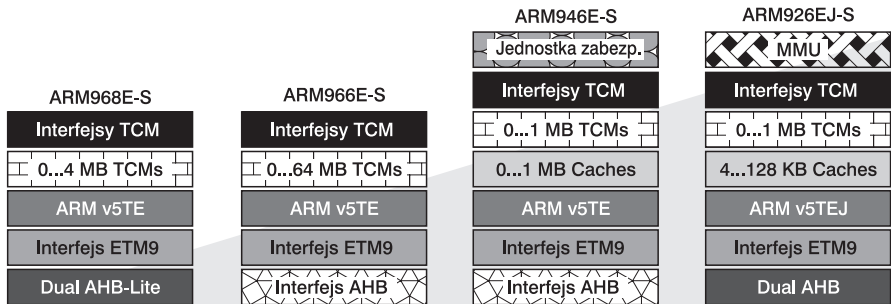
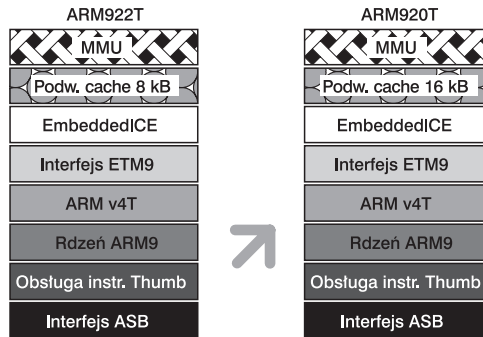
Rys. 1.2. Rodzina rdzeni ARM7

Tab. 1.1. Zestawienie list rozkazów i wyposażenia wybranych architektur rdzeni ARM

Rdzeń	Architektura	Lista rozkazów
ARM7TDMI	ARMv4T	Thumb
ARM9	ARMv5TE	ARM, Thumb
ARM9E	ARMv5TEJ	ARM, Thumb, DSP, Jazelle
ARM10	ARMv5TEJ	ARM, Thumb, DSP, Jazelle
ARM11	ARMv6	Thumb, Thumb-2
Cortex	ARMv7	Thumb-2

wybrane warianty zestawiono w **tabeli 1.1**. Trzeba więc pamiętać, że ARM v7 (oznaczenie architektury) nie oznacza tego samego co ARM7 (oznaczenie rodziny rdzeni)! Co więcej, inżynierowie firmy ARM Holdings plc stosują w ramach rodziny ARM7 architektury ARM v4T (ARM7TDMI, ARM7TDMI-S oraz ARM720T) oraz ARM v5TEJ (ARM7EJ-S) – **rysunek 1.2**. Podobny problem występuje w rodzinie rdzeni ARM9E (**rysunek 1.3**), ale uniknięto go w rodzinie ARM9 (**rysunek 1.4**).

Kolejną, tym razem uporządkowaną, lecz nie zawsze oczywistą sprawą są obsługiwane przez rdzenie listy rozkazów. Pierwsze rdzenie opracowane przez firmę ARM

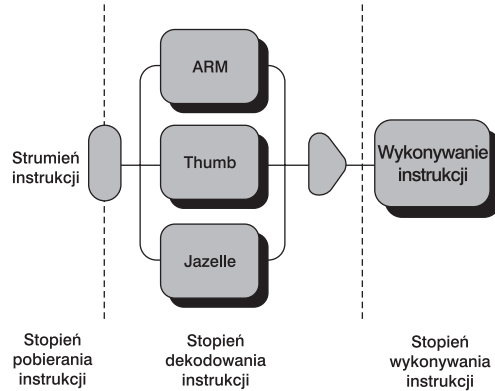
**Rys. 1.3.** Rodzina rdzeni ARM9E**Rys. 1.4.** Rodzina rdzeni ARM9

Holdings plc były przystosowane do korzystania z 32-bitowych poleceń (co należy rozumieć, że szerokość słowa rozkazowego miała 32 bity). Dość szybko okazało się, że nie jest to rozwiązanie ekonomiczne – pamięci Flash o szerokości słowa wynoszącej 32 bity są rzadko spotykane, natomiast „składanie” takich słów z dwóch 16-bitowych lub czterech 8-bitowych poważnie odbija się na prędkości wykonywania programu. Rozwiązaniem tego problemu jest zastosowany po raz pierwszy w rdzeniach ARM7 zmodyfikowany zestaw poleceń o 16-bitowej szerokości słowa – nosi on nazwę Thumb. Thumb w nomenklaturze firmy ARM Holdings plc oznacza, że procesor może wykonywać program z pamięci o szerokości magistrali 8 lub 16 bitów, podczas gdy standardowa szerokość słowa danych wynosi 32 bity. Każde 32-bitowe polecenie jest „kompresowane” do postaci 16-bitowej i po odczytaniu z pamięci programu wykonywane w taki sam sposób jak polecenie „oryginalne”. Dzięki takiemu rozwiązaniu znacznie maleje koszt wykonania systemu, zwłaszcza w przypadku korzystania z zewnętrznej pamięci programu.

Lista instrukcji dostępnych w wersji Thumb jest krótsza niż na pierwotnej liście poleceń 32-bitowych, ale obejmuje polecenia najczęściej stosowane w programach, co nieco ogranicza programistów (a raczej kompilatory języków wyższego poziomu), ale pozwala korzystać z tanich pamięci Flash o 16-bitowym słowie. Jak łatwo zauważyć, w przypadku korzystania z listy rozkazów Thumb znacznie mniej kosztowne (czasowo) jest korzystanie z pamięci 8-bitowych – pobranie kompletnego rozkazu wymaga tylko dwóch, a nie czterech (jak w przypadku klasycznych poleceń ARM) odczytów z pamięci programu. Ważne jest, żeby nie traktować poleceń Thumb jako prostego podzbioru instrukcji ARM.

Polecenia na liście Thumb są w większości przypadków równie efektywne, jak ich klasyczne odpowiedniki, ale zmodyfikowany system ich binarnego kodowania (producent wspomina nawet o kompresji słów poleceń) powoduje, że uzyskano poważne zmniejszenie długości programów (nawet do 30%, co przenosi się m.in. na wypadkowy pobór mocy przez system) przy relatywnie niewielkim spadku prędkości ich wykonywania. Lista Thumb w praktyce sprawdziła się na tyle nieźle, że radzą sobie z nią wszystkie współcześnie dostępne mikrokontrolery z rdzeniami ARM, ale producent cały czas prowadził „w tle” eksperymenty z nową listą poleceń – Thumb-2. Jej idea jest praktycznie identyczna jak w przypadku Thumb, ale uwzględniono w niej wymogi współczesności: listę rozkazów rozszerzono o instrukcje wspomagające aplikacje DSP (tzw. SIMD-ARM, SIMD – *Single Instruction Multiple Data*), dodano nowe instrukcje zgodne z „klasyczną” listą ARM, a także nowe rozkazy 16-bitowe ułatwiające organizację działania programu (m.in. kompleksowe rozkazy działania na bitach, a także assemblerowe polecenia odpowiadające instrukcji warunkowej *If-Then*).

Mikroprocesory z rdzeniami ARM podbiły rynek aplikacji przenośnych, w tym – a jest to jeden z największych rynków – telefonię GSM i UMTS. Wymagania stawiane nowoczesnym telefonom komórkowym już dawno sięgnęły multimediiów, co postawiło bardzo poważne zadania (chodzi oczywiście o maksymalną moc obliczeniową uzyskiwaną przy jak najmniejszym poborze mocy) przed konstruktorami telefonów. Trendu nie przeoczyli inżynierowie z firmy ARM i niektóre z oferowanych przez siebie rdzeni wyposażyli w dodatkowy zestaw instrukcji (inicjowany poleceniem *Branch-to-Java*) ułatwiający szybkie wykonywanie aplikacji napisanych w Javie.



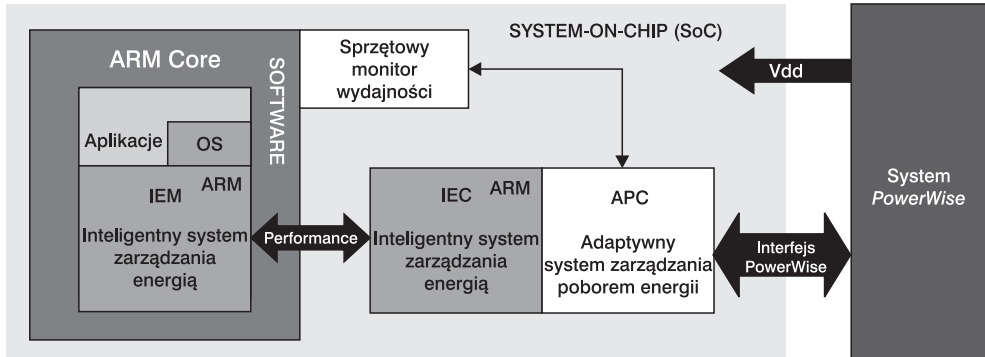
Rys. 1.5. Rdzenie z sufiksem „J” wyposażono w dodatkową ścieżkę do wykonywania poleceń wspomagających pracę interpretera Javy (moduł Jazelle)

Prezentowane rozszerzenie sprzętowe rdzeni producent nazwał Jazelle DBX (*Data Byte eXecution*). Rozwiązaniem alternatywnym do rdzenia ze sprzętowym modułem Jazelle jest oprogramowanie VTK przygotowane przez firmę ARM Holdings plc dla starszych architektur: ARMv4, ARMv4T oraz ARMv5TE (**rysunek 1.5**), które umożliwia szybszą interpretację programów napisanych w Javie niż ma to miejsce w rozwiązaniach klasycznych.

Niektóre rdzenie z ARM Holdings plc są oferowane z rozszerzeniem listy rozkazów o nazwie NEON. Są to polecenia umożliwiające między innymi równoległe obliczenia z klasy DSP na 16 próbkach sygnału jednocześnie, co powoduje, że możliwe staje się: sprzętowe obliczanie FFT w czasie rzeczywistym, strumieniowe kodowanie/dekodowanie MPEG4, a także rozpoznawanie w czasie rzeczywistym ręcznego pisma i odcisków palców.

Firma ARM Holdings plc oferuje ponadto sprzętowe rozszerzenia rdzeni ARM9, ARM10E oraz ARM11, których zadaniem jest wspomaganie wyświetlania grafiki trójwymiarowej (*MBX 3D Accelerator*) oraz sprzętowy rendering (*Swerve Client*). Deklarowane wydajności uzyskiwane przez „mikromocowe” rdzenie ARM pozwalają uruchamiać na nich wszelkie współczesne gry komputerowe – jest to rekomendacja najlepsza z możliwych, od lat bowiem to właśnie gry są motorem napędowym techniki mikroprocesorowej. Kolejną możliwością oferowaną twórcom urządzeń multimedialnych są rdzenie wyposażone w rozszerzenie listy poleceń o takie, które są najczęściej stosowane w aplikacjach DSP. Obecnie rozszerzenie ARM-DSP jest dostępne w rdzeniach ARMv5TE oraz ARMv6.

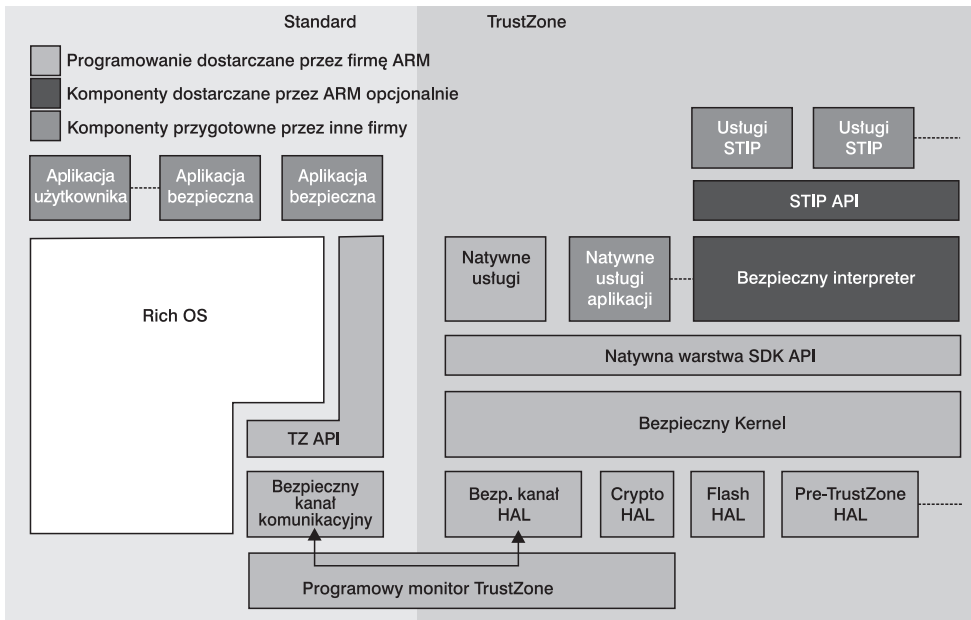
Budowa rdzeni oferowanych przez ARM Holdings plc jest maksymalnie uproszczona, dzięki czemu pobierana przez nie moc jest stosunkowo niewielka. Praktyka pokazała jednak, że konieczne jest zastosowanie dodatkowych mechanizmów oszczędnościowych, pośród których najczęściej jest stosowany system IEM (*Intelligent Energy Manager*). Jest to zintegrowany system sprzętowo-programowy wykorzystujący blok IEC (*Intelligent Energy Controller*), który współpracuje z opracowanymi przez firmę National Semiconductor układami zasilającymi PowerWise (**rysunek 1.6**). Jednym z zadań IEM jest analiza wymagań aplikacji i adekwatna do nich zmiana napięcia zasilającego rdzeń (*Adaptive Voltage Scaling*). Rozwiązanie to należy potraktować



Rys. 1.6. Energooszczędność rdzeni ARM wspomagają zaawansowane systemy zasilania, jak na przykład PowerWise firmy National Semiconductor

jako ciekawostkę – nie jest ono bowiem na razie dostępne w produkowanych mikrokontrolerach, mogą z niego korzystać wyłącznie twórcy układów ASIC.

Kolejnym segmentem rynku, bardzo poważnie potraktowanym przez firmę ARM, są aplikacje wymagające zapewnienia bezpieczeństwa dostępu do danych. Pierwszym wprowadzonym na rynek rdzeniem z mechanizmami TrustZone (**rysunek 1.7**) był ARM1176JZ(F)-S, jak na razie dostępny wyłącznie w postaci syntezywalnego rdzenia dla układów SoC. Ponieważ rozwiązania wyłącznie sprzętowe nie są wystarczająco niezawodne, ARM Holdings plc nawiązała współpracę z firmą Trusted Logic, która przygotowuje środowiska programowe zapewniające uzyskanie wysokiego współczynnika bezpieczeństwa aplikacji, w tym dużej odporności na ataki kryptograficzne.



Rys. 1.7. Schemat blokowy aplikacji wykorzystującej mechanizmy ochronne TrustZone

Rdzenie ARM są optymalizowane pod kątem wydajności i poboru mocy, ale każda podrodzina wyróżnia się indywidualnymi cechami (obsługiwany zestaw instrukcji, wspomaganie DSP i Javy, możliwość sterowania poborem mocy itp.), dość ściśle określającymi docelowe obszary ich zastosowań. Na szczęście, jednym z założeń twórców rdzeni ARM było zapewnienie ich maksymalnej wzajemnej kompatybilności. Utrzymywana jest także kompatybilność „w dół”, dzięki czemu programy wcześniej przygotowane mogą być stosowane także w systemach wykorzystujących procesory ARM nowszych generacji. Najczęściej spotykane w popularnych urządzeniach rdzenie ARM to: ARM7, ARM9/ARM9E oraz Cortex.

1.1. Rdzeń ARM7

Łatwo oszacować, że wydajność procesora z 32-bitowym RISC-owym rdzeniem jest na tyle duża, że do wykorzystania jego wszystkich możliwości konieczna jest niebanalna aplikacja. Dodatkowo ostrożność wśród konstruktorów wzbudzał fakt, że procesory 32-bitowe były bardzo kosztowne i wymagały kosztownego otoczenia. Trudne było także projektowanie płytek drukowanych, przede wszystkim ze względu na konieczność prowadzenia magistral o dużej szerokości.

Dość szybko po rozpoczęciu ekspansji układów z rdzeniem ARM na rynku okazało się, że jest duże zapotrzebowanie na tani i energooszczędny procesor 32-bitowy o względnie dużej szybkości. Z myślą o zaspokojeniu takich potrzeb ARM wprowadził do swojej oferty rdzenie ARM7, które są dostępne w trzech wariantach: ARM7TDMI/ARM7TDMI-S, ARM7EJ i ARM720T. Maksymalna prędkość tych rdzeni wynosi 120 MIPS (typowo 0,9...1 MIPS/MHz), natomiast możliwy do uzyskania niewielki pobór mocy (typowe, krzemowe implementacje zadowolają się poborem mocy z zakresu 0,06...0,25 mW/MHz) umożliwia stosowanie ich w urządzeniach przenośnych zasilanych bateryjnie. Między innymi te parametry zadecydowały o tym, że rdzenie ARM7 są często wykorzystywane przez producentów mikrokontrolerów. W układach tych rdzeń ARM7 jest otoczony różnorodnymi blokami peryferyjnymi, podobnie jak to się dzieje w popularnych mikrokontrolerach (np. z rodziny AVR).

Popularność ARM7 wynika także ze wsparcia ze strony producentów systemów operacyjnych i łatwości przeniesienia aplikacji przygotowanej dla ARM7 na rdzenie nowszych generacji: ARM9 i ARM10. Dostępne są zarówno komercyjne systemy operacyjne (jak Windows CE, Symbian, Palm OS czy Portable Linux), jak i rozwiązania bezpłatne (np. ARM Linux, ARMed OS).

1.2. Rdzeń ARM9/ARM9E

Architektura rdzeni ARM9 jest bardzo zbliżona do ARM7, z tą różnicą, że ARM9 są bezkompromisowo zorientowane na maksymalizację wydajności. Spowodowało to wzrost poboru mocy do 0,8 mW/MHz, a maksymalna częstotliwość taktowania dochodzi do 250 MHz (ARM920T). Wzrost wydajności uzyskano m.in. dzięki wyposażeniu rdzenia w pamięci cache dla instrukcji i danych (4...16 kB), które współpracują z blokami zarządzającymi ich pracą MMU (*Memory Management Unit*), chociaż nie wszystkie krzemowe mikrokontrolery i mikroprocesory z rdzeniem

ARM9 są w te bloki wyposażane. Rdzenie ARM9E wyposażono ponadto w interfejs przyspieszający dostęp do pamięci TCM (*Tightly Coupled Memory*), a ich architektura jest zgodna ze specyfikacją ARMv5TE/TEJ zamiast ARMv4T, która jest standardowa dla ARM9.

1.3. Rdzenie Cortex

Rodzina rdzeni Cortex składa się z trzech podrodzin:

- Cortex-A – układy przeznaczone do stosowania w systemach wymagających dużej wydajności obliczeniowej, w tym multimedialnych;
- Cortex-R – układy przeznaczone do stosowania w systemach czasu rzeczywistego;
- Cortex-M – układy przeznaczone do stosowania w mikrokontrolerach.

Rdzenie Cortex są zgodne z Thumb-2 (niektóre wersje obsługują tylko ten sposób kodowania rozkazów, bez możliwości korzystania z zestawu klasycznych instrukcji ARM oraz Thumb) i mają one nową w ofercie ARM Holdings plc architekturę oznaczoną symbolem ARM v7A/R/M. Atutami rdzeni Cortex są: niski koszt implementacji w krzemie dzięki niewielkiej powierzchni wynoszącej 0,56 mm² (dzięki dostosowaniu do technologii 0,18 μm oraz zoptymalizowaniu budowy rdzenia), wysokie dopuszczalne częstotliwości taktowania, a także zaawansowane mechanizmy przyspieszania pracy rdzenia (m.in. sterowane przepływem danych w pamięci cache z wielostopniową predykcją, skrócenie czasu obsługi przerwań, zastosowanie sterownika TCM).

1.4. Zintegrowane debugowanie

Trudnym do przecenienia atutem procesorów z rdzeniem ARM jest wyposażenie ich w elementy ułatwiające monitorowanie i debugowanie ich pracy. Najważniejszym mechanizmem jest Embedded ICE-RT, umożliwiający pogląd stanów rejestrów i pamięci, a także ustawianie pułapek (zatrzymujących i monitorujących). Dostęp do tych funkcji jest możliwy poprzez interfejs JTAG z klasycznym automatem TAP na wejściu.

Opcjonalnym wyposażeniem niektórych procesorów ARM jest blok ETM (*Enhanced Trace Macrocell*), umożliwiający monitorowanie stanu wewnętrznych magistral w czasie rzeczywistym. Jest on przydatny szczególnie wtedy, gdy procesor jest implementowany w układzie SoC, w którym komunikuje się z niedostępnymi na zewnątrz blokami peryferyjnymi. Dostęp do informacji z ETM jest możliwy poprzez specjalny interfejs (*Trace Port*) lub JTAG – dane są gromadzone w specjalnej pamięci buforującej ETB (*Embedded Trace Buffer*).

Zalety ETM i współpracującego z nim interfejsu JTAG niweczy w pewnym stopniu duża liczba wyprowadzeń niezbędnych do korzystania z możliwości sprzętowego debuggera. Z tego powodu ARM opracował zmodyfikowaną wersję interfejsu o nazwie SWD (*Serial Wire Debug*), który zamiast 4- lub 5-liniowego JTAG-a wykorzystuje interfejs 2-przewodowy.