

Systemy wbudowane

Porównanie mikrokontrolerów AVR, ARM z
procesorami rodziny x86/64

Procesory ARM

- Jeden z pierwszych procesorów w pełni 32 bitowych
- Układ tworzony w całości od podstaw, bez narzutów historycznych poprzednich generacji
- Implementacja oparta na rozwiązaniu RISC (nie jest to ideowa implementacja!)
- W pełni zoptymalizowane długości rozkazów (jeden cykl => jeden takt zegara procesora)
- Architektura pozwoliła na redukcję tranzystorów na płycie krzemowej, co z kolei przekłada się na większą ilość rejestrów dostępnych w procesorze

Procesory ARM

- Wartości operacji w rejestrach ARM nie są cyklicznie czyszczone; przyspiesza to znacznie często powtarzane operacje
- Niestety architektura ARM posiada specyficzne wymagania stawiane inżynierom oprogramowania – aplikacje pisane pod nie muszą być optymalizowane!
- Wstępnie procesory posiadały problem gęstości instrukcji; większość rozkazów była 16 bitowa, jednak pojedynczy rozkaz musi być zawsze jednym słowem (32 bit)

Ogólna charakterystyka ARM

- Duży, jednolity plik rejestru
- Ładowanie/przechowywanie danych w rejestrze zapobiega bezpośredniemu dostępowi do pamięci
- Adresacja ładowania/przechowywania uzależniona wyłącznie od instrukcji i zawartości rejestrów
- Jednolita wielkość pola instrukcji
- Nadzór nad jednostką ALU

Ogólna charakterystyka ARM

- Automagiczne zwiększanie i zmniejszanie trybów adresacji celem optymalizacji pętli programu
- Ładowanie i przechowywanie wielu instrukcji w celu zwiększania przepływności danych
- warunkowe wykonywanie instrukcji programu w celu zwiększenia wydajności ich wykonywania

Rozwiązania dostępne w ARM

- Gęstość kodu rozwiązana przez mechanizm Thumb
- Do piątej generacji procesory miały pamięć organizowaną wedle architektury von Neumann
- Architektura Harvard zwiększyła przepustowość magistrali danych, jednak dopiero 64 bitowa szyna (ARMv6) wyeliminowała tzw. wąskie gardło

Rejestry ARM

- Procesory posiadają łącznie 37 rejestrów, z czego 20 jest widocznych dla programisty
- 31 rejestrów ma status ogólnego przeznaczenia, pozostałe to rejestry stanu procesora
- Dostęp do rejestrów definiowany jest przez tryb pracy procesora (przypomina uprawnienia aplikacji)

Rejesty ARM

User and system

<i>r0</i>					
<i>r1</i>					
<i>r2</i>					
<i>r3</i>					
<i>r4</i>					
<i>r5</i>					
<i>r6</i>					
<i>r7</i>					
<i>r8</i>	<i>r8_fiq</i>				
<i>r9</i>	<i>r9_fiq</i>				
<i>r10</i>	<i>r10_fiq</i>				
<i>r11</i>	<i>r11_fiq</i>				
<i>r12</i>	<i>r12_fiq</i>				
<i>r13 sp</i>	<i>r13_fiq</i>	<i>r13_irq</i>	<i>r13_svc</i>	<i>r13_undef</i>	<i>r13_abt</i>
<i>r14 lr</i>	<i>r14_fiq</i>	<i>r14_irq</i>	<i>r14_svc</i>	<i>r14_undef</i>	<i>r14_abt</i>
<i>r15 pc</i>					
<i>cpsr</i>					
-	<i>spsr_fiq</i>	<i>spsr_irq</i>	<i>spsr_svc</i>	<i>spsr_undef</i>	<i>spsr_abt</i>

Fast interrupt request

Interrupt request

Supervisor

Undefined

Abort

Rejestr aktualnego stanu programu

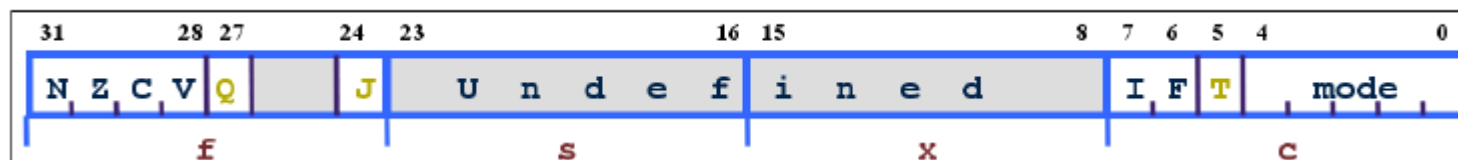
Flagi stanu kodu

- N – Negatywny wynik od ALU
- Z – wynik Zero od ALU
- C – operacja przeniesienia (Carried)
- V – informacja o przepełnieniu (oVerflow)
- Flaga lepkiego przepełnienia – oznaczana jako Q – informuje, że miało miejsce nasycenie kodu (tylko architektura 5TE/J)
- Flaga informująca o pracy układu w trybie Jazelle (tryb wspomagający wykonywanie kodu bajtowego Java) – oznaczana jako J – procesor pracuje w tym trybie, gdy wartość flagi wynosi 1 (tylko architektura 5TEJ)

Flagi wyłączenia przerwań:

- I = 1 – wyłączone IRQ
- F = 1 – wyłączone FIQ
- Flaga informująca o budowie instrukcji – oznaczona jako T (tylko architektura xT):
 - T = 0 – procesor działa w trybie ARM
 - T = 1 – procesor działa w trybie Thumb

Bity trybu – definiują tryb procesora.



Tryby pracy ARM

Mode	Description	
Supervisor (SVC)	Entered on reset and when a Software Interrupt instruction (SWI) is executed	Privileged modes
FIQ	Entered when a high priority (fast) interrupt is raised	
IRQ	Entered when a low priority (normal) interrupt is raised	
Abort	Used to handle memory access violations	
Undef	Used to handle undefined instructions	
System	Privileged mode using the same registers as User mode	Unprivileged mode
User	Mode under which most Applications / OS tasks run	

Exception modes

Schemat nazewnictwa ARM

ARM {x} {y} {z} {T} {D} {M} {I} {E} {J} {F} {S}

- x – rodzina (wersja)
- y – ilość pamięci zarządzanej/chronionej
- z – ilość pamięci cache
- T – tryb Thumb
- D – odpluskwiacz JTAG (standardowy protokół szeregowy odbioru i wysyłania informacji pomiędzy mikroprocesorem a zestawem testującym)
- M – szybki mnożnik
- I – EmbeddedICE macrocell (umożliwia odpluskwianie poprzez ustawianie punktów przerwań oraz punktów obserwacyjnych)
- E – ulepszone instrukcje (obejmujące TDMI)
- J – tryb Jazelle
- F – jednostka zmiennoprzecinkowych wektorów
- S – wersja synchroniczna

Procesory x86_64

- Jedna z najstarszych nadal używanych architektur procesorów
- Jej jeden z najstarszych przedstawicieli (8008) nadal może być spotkany w najtańszych kalkulatorach
- Do niedawna (o ile nie nadal) NASA wykorzystywała model 8086/88 do obliczeń w sprzęcie kosmicznym

Procesory x86_64

- Opracowane pierwotnie jako CISC
- Obecnie stanowią hybrydę pomiędzy zestawami CISC i RISC – rozkazy CISC zamieniane są na mikrokod
- Nowsze procesory są zgodne ze starymi wersjami; oznacza to, że teoretycznie nawet najstarsze aplikacje powinny działać na nowych procesorach tej rodziny (TEORETYCZNIE!)

Charakterystyka x86_64

- Przestrzeń adresowa – obecnie 2^{64}
- Najmniejsza adresowalna komórka – 8 bit
- 8 rejestrów ogólnego przeznaczenia
- Ponadto, w najnowszych wersjach procesorów można spotkać także rejestry SSE (8x128 bit oraz 16x256 bit), rejestry MMX (8x64 bit)
- Implementacja liczb całkowitych (8,16,32 oraz 64 bitowe)
- Implementacja liczb zmiennoprzecinkowych (32 oraz 64 bitowe)
- Wektory liczb całkowitych/zmiennoprzecinkowych 64,128 oraz 256 bitowe

Charakterystyka x86_64

- Typ całkowity (integer) dzieli się ponadto na znakowy, krótki, całkowity oraz długi całkowity; typ może być znakowy oraz bezznakowy
- Zmiennoprzecinkowy rozróżniany jest na 32 bitowy (float) oraz 64 bitowy podwójnej precyzji (double)
- Wektorowe typy umożliwiają wykonywanie operacji na większych danych przez rejestry MMX czy SSE. Ich zastosowanie podyktowane jest pod rozwiązaniami multimedialne

Charakterystyka x86_64

- Typ całkowity (integer) dzieli się ponadto na znakowy, krótki, całkowity oraz długi całkowity; typ może być znakowy oraz bezznakowy
- Zmiennoprzecinkowy rozróżniany jest na 32 bitowy (float) oraz 64 bitowy podwójnej precyzji (double)
- Wektorowe typy umożliwiają wykonywanie operacji na większych danych przez rejestry MMX czy SSE. Ich zastosowanie podyktowane jest pod rozwiązaniami multimedialne

Rejestry x86_64

4 rejestry ogólnego przeznaczenia:

- R(E)AX – akumulator
- R(E)BX – baza
- R(E)CX – licznik pętli
- R(E)DX – rozszerzenie licznika pętli

Indeksowe i wskaźnikowe:

- R(E)SI – indeks źródła dla operacji na łańcuchach
- R(E)DI – indeks celu dla operacji na łańcuchach
- R(E)SP – rejestr stosu dla najwyższej wartości stosu
- R(E)BP – wskaźnik bazowy aktualnej ramki stosu
- R(E)IP – wskaźnik instrukcji

Segmenty x86_64

- CS – segment kodu
- DS – segment danych
- SS – segment stosu
- ES – dodatkowy segment danych
- FS – dodatkowy segment danych
- GS – dodatkowy segment danych

Struktura rejestrów

General Purpose Registers (A, B, C and D)

64	56	48	40	32	24	16	8
R?X							
				E?X			
						?X	
						?H	?L

64-bit mode-only General Purpose Registers (R8, R9, R10, R11, R12, R13, R14, R15)

64	56	48	40	32	24	16	8
?							
				?D			
						?W	
						?B	

Struktura rejestrów

Segment Registers (C,
D, S, E, F and G)

16	8
?S	

Pointer Registers (S and B)

64	56	48	40	32	24	16	8
R?P							
				E?P			
						?P	
							?PL

Note: The ?PL registers are only available in 64-bit mode.

Index Registers (S and D)

64	56	48	40	32	24	16	8
R?I							
				E?I			
						?I	
							?IL

Note: The ?IL registers are only available in 64-bit mode.

Instruction Pointer Register 

64	56	48	40	32	24	16	8
RIP							
				EIP			
						IP	

Podsumowanie x86_64

- Procesory te nie nadają się do stosowania w rozwiązaniach mikroukładów
- Głównym problemem jest tutaj wysokie zapotrzebowanie na energię
- Intel początkowo próbował implementacji ARM (StrongARM), później stworzył własną platformę przenośną – Atom (Centrino Atom)

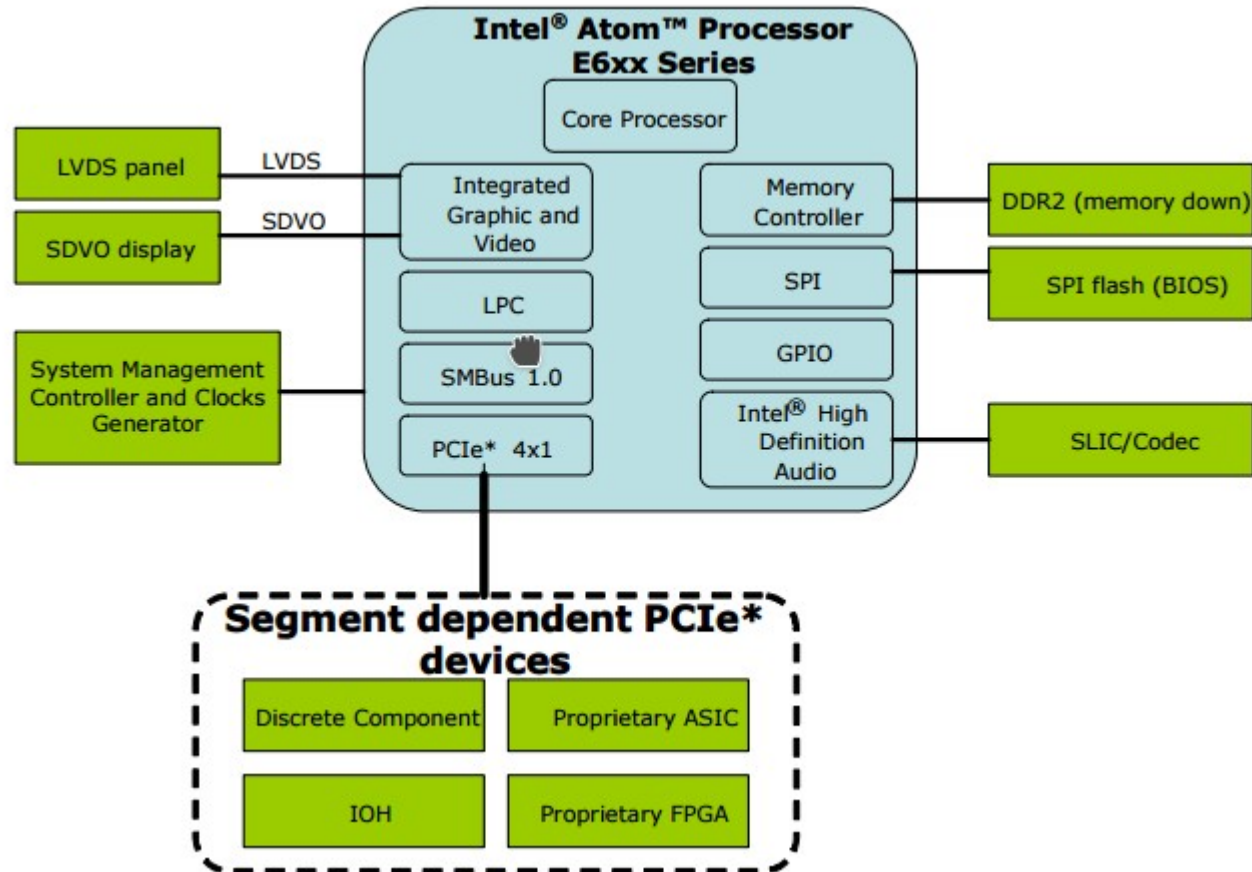
Układy Intel Atom

- Zaprojektowane głównie z myślą o rynku przenośnym
- Pierwsze układy były niezwykle wydajne – pobierały zaledwie 3 W energii przy zegarze 800 MHz
- Niestety procesory te są jedynie zgodne z architekturą x86 (należą do grupy ultra low voltage IA32)
- Pierwotne układy nie posiadały wsparcia architektury 64 bit (obecne posiadają)

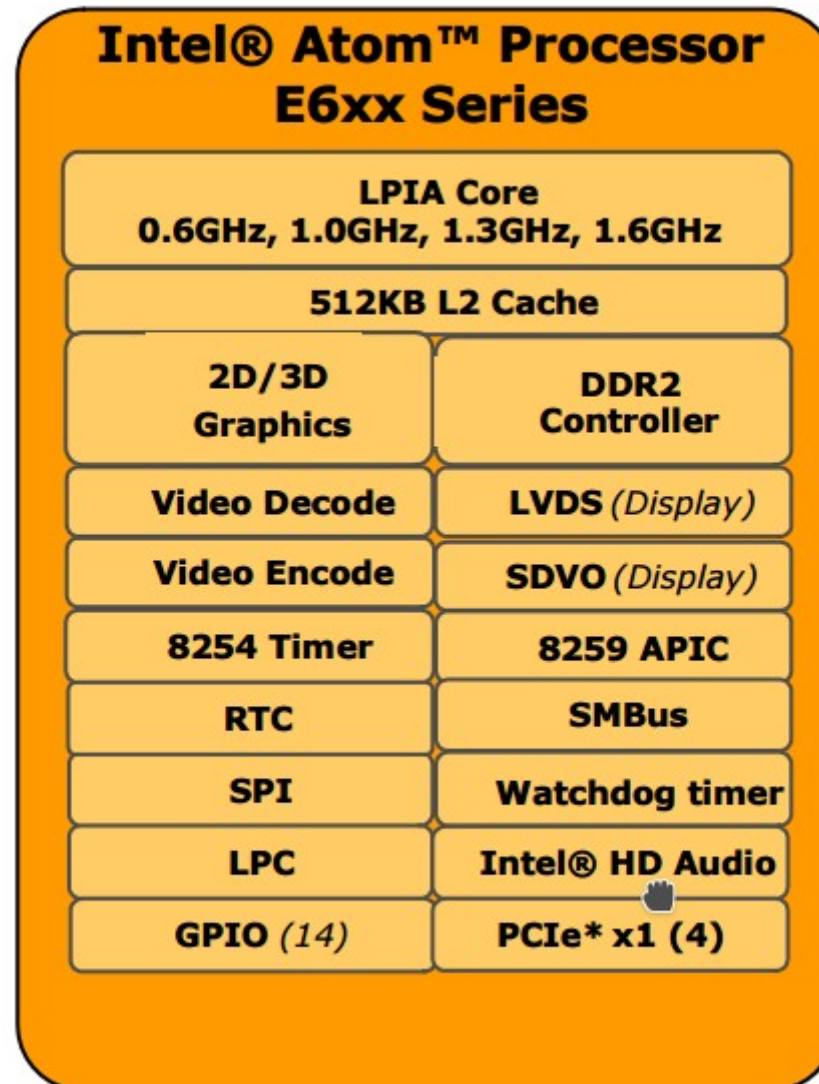
Układy Intel Atom

- Nowsze układy Atom zostały zbudowane w oparciu o architekturę wielordzeniową (pierwotne wersje nie posiadały nawet wsparcia wielowątkowości)
- Im większe taktowanie i możliwości, tym większe zapotrzebowanie na energię (najnowsze edycje potrzebują nawet 20 W)
- Układ zaimplementowany został w kilkudziesięciu urządzeniach, w tym mikrokomputerze (konkurencja dla RaspberryPi 3), telefonach (Asus ZenFone), czy urządzeniach do internetu (MID).

Schemat układu Intel Atom



Układ Intel Atom



Materialy - ARM

- <http://llvm.org/devmtg/2012-11/Northover-AArch64.pdf>
- https://www.arm.com/files/downloads/ARMv8_white_paper_v5.pdf
- <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.subset.architecture.reference/index.html>
- <https://www.arm.com/products/processors/armv8-architecture.php>
- https://static.docs.arm.com/ddi0487/a/DDI0487A_k_armv8_arm_iss10775.pdf
- Qualcomm® Snapdragon™ 600E Processor APQ8064E Data Sheet, September 2016

Charakterystyka x86_64

- Typ całkowity (integer) dzieli się ponadto na znakowy, krótki, całkowity oraz długi całkowity; typ może być znakowy oraz bezznakowy
- Zmiennoprzecinkowy rozróżniany jest na 32 bitowy (float) oraz 64 bitowy podwójnej precyzji (double)
- Wektorowe typy umożliwiają wykonywanie operacji na większych danych przez rejestry MMX czy SSE. Ich zastosowanie podyktowane jest pod rozwiązaniami multimedialne

Materialy - AVR

- http://www.atmel.com/Images/Atmel-2486-8-bit-AVR-microcontroller-ATmega8_L_datasheet.pdf
- <http://www.atmel.com/images/doc2466.pdf>
- <http://www.atmel.com/images/doc32000.pdf>
-

Materialy – Intel/AMD

- <http://www.hardwaresecrets.com/inside-atom-architecture/>
- <https://www.cis.upenn.edu/~cdmurphy/cit593/fall2012/pdf/lecture20-x86.pdf>
- https://en.wikipedia.org/wiki/Intel_Atom
- <https://pl.wikipedia.org/wiki/X86-64>
- <http://www.tomshardware.com/reviews/intel-larrabee-graphics,2253-6.html>
- <https://software.intel.com/en-us/node/540484>
- <https://en.wikipedia.org/wiki/X86>
- <http://www.computerworld.com/article/2535019/computer-hardware/timeline--a-brief-history-of-the-x86-microprocessor.html>
- <http://docs-europe.electrocomponents.com/webdocs/14a9/0900766b814a9f55.pdf>