

System wbudowany (ang. *embedded system*) – system komputerowy specjalnego przeznaczenia, który staje się integralną częścią obsługiwanego przez niego sprzętu komputerowego (*hardware*). (Wikipedia)

System wbudowany musi spełniać określone wymagania ściśle zdefiniowane pod kątem zadań, które ma wykonywać. Nie można więc nim nazywać typowego wielofunkcyjnego komputera osobistego. Każdy system wbudowany oparty jest na mikroprocesorze (lub mikrokontrolerze) zaprogramowanym do wykonywania ograniczonej liczby zadań lub nawet wyłącznie do jednego.

Zależnie od przeznaczenia może zawierać oprogramowanie przeznaczone jedynie na to urządzenie (firmware) lub system operacyjny wraz ze specjalizowanym oprogramowaniem. Zwykle decyduje o tym stopień niezawodności, jaki ma oferować dany system wbudowany. Ogólną zasadą jest, że im mniej złożone i specjalizowane jest oprogramowanie, tym bardziej system jest niezawodny i pozwala szybciej reagować na zdarzenia krytyczne.

Niezawodność systemu może być zwiększona poprzez rozdzielenie zadań na mniejsze podsystemy, a także przez redundancję. Może ona polegać na zastosowaniu do jednego zadania dwóch identycznych urządzeń, z których jedno przejmuje zadania drugiego w przypadku jego awarii.

Za pierwszy komputer wbudowany jest uznawany ten sterujący amerykańskim statkiem kosmicznym Apollo. Natomiast pierwszy komputer wbudowany produkowany masowo służył do sterowania rakieta LGM-30 Minuteman I.

Zastosowania

Obecnie systemy wbudowane znajdują zastosowanie praktycznie we wszystkich dziedzinach, gdyż ogólnie dąży się do tego, aby wszystkie urządzenia były tzw. „inteligentne” i zdolne do pracy autonomicznej, a także mogły wykonywać coraz to bardziej złożone zadania.

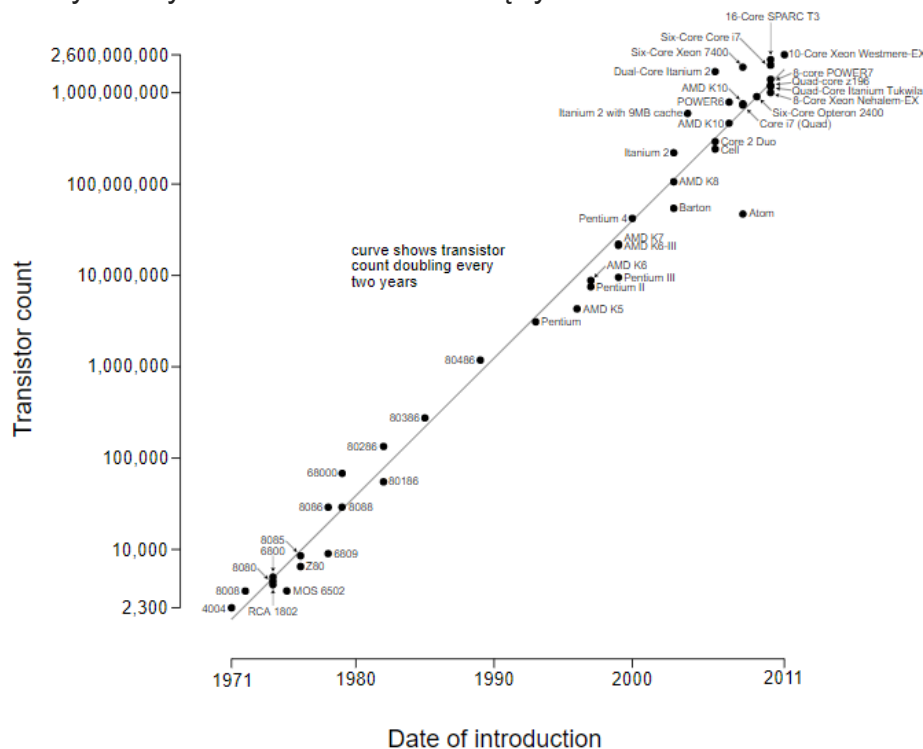
Przykłady obszarów i urządzeń technicznych, w których stosuje się systemy wbudowane:

- układy sterujące pracą silnika samochodowego i ABS, komputery pokładowe;
- sprzęt sterujący samolotami, raketami, pociskami raketowymi, inteligentnymi bombami;
- sprzęt medyczny, w tym m.in. monitory holterowskie;
- sprzęt pomiarowy, w tym m.in. oscyloskopy, analizatory widma;
- bankomaty i podobne urządzenia ATM;
- termostaty, klimatyzatory;
- kuchenki mikrofalowe, zmywarki;
- sterowniki PLC stosowane w przemyśle do sterowania, kontroli procesów i maszyn produkcyjnych;
- sterowniki do wszelkiego rodzaju robotów mechanicznych;
- systemy alarmowe służące do ochrony osób i mienia np. antywłamaniowe, przeciwpożarowe i inne;
- telefony komórkowe i centrale telefoniczne;
- drukarki, kserokopiarki;
- kalkulatory;

- sprzęt komputerowy, w tym między innymi: dyski twarde, napędy optyczne, routery, serwery czasu i firewalle;
- systemy rozrywki multimedialnej i interaktywnej:
 - konsole do gier, stacjonarne i mobilne;
 - automaty do gier oraz o innym zastosowaniu;
 - telewizory, odtwarzacze DVD, kamery cyfrowe, magnetowidy, set-top boxy.

Prawo Moore'a

Gordon Moore, jeden z założycieli firmy Intel, w 1965 r. zaobserwował podwajanie się liczby tranzystorów co ok. 18 miesięcy



Litografia i liczba tranzystorów w przykładowych procesorach na przestrzeni lat:

- 10 μm – 1971 rok, Intel 4004; liczba tranzystorów - 2,300
- 6 μm – 1974 rok, Motorola 6800; liczba tranzystorów - 4,100
- 3 μm – 1976 rok, Intel 8085; liczba tranzystorów - 6,500
- 1.5 μm – 1982 rok, Intel 80286; liczba tranzystorów - 134,000
- 800 nm – 1987 rok, Motorola 68030; liczba tranzystorów - 273,000
- 350 nm – 1996 rok, AMD K5; liczba tranzystorów - 4,300,000
- 180 nm – 1999 rok, Pentium II Mobile Dixon; liczba tranzystorów - 27,400,000
- 90 nm – 2004 rok, Pentium 4 Prescott; liczba tranzystorów - 112,000,000
- 65 nm – 2007 rok, AMD K10 quad-core 2M L3; liczba tranzystorów - 463,000,000
- 32 nm – 2010 rok, Core i7 (Gulftown); liczba tranzystorów - 1,170,000,000
- 20 nm – 2014 rok, Apple A8; liczba tranzystorów - 2,000,000,000
- 10 nm – 2016 rok, Qualcomm Snapdragon 835; liczba tranzystorów - 3,000,000,000

Według dokumentów *International Technology Roadmap for Semiconductors*, uwzględniających potencjalne problemy z rozwojem i miniaturyzacją, należy oczekiwać kolejnych procesów dostępnych (na rynku) w latach, 32 nm – 2009, 22 nm – 2012, 14 nm – 2015, 10 nm – 2018(Intel), 7 nm - 2019(AMD), 5nm ~ 2020, 3nm ~ 2022, a dalszy rozwój w ramach elektroniki stoi pod znakiem zapytania. Ostatnie 10 lat będą miały mniejszą dynamikę wzrostu niż wskazuje na to Prawo Moore'a.

Mikrokontroler (MCU, μC , ang. *microcontroller*), mikrokomputer jednoukładowy – scalony system mikroprocesorowy, zrealizowany w postaci pojedynczego układu scalonego zawierającego jednostkę centralną (CPU), pamięć RAM oraz rozbudowane układy wejścia-wyjścia i na ogół pamięć programu jako ROM lub Flash.

Określenie *mikrokontroler* pochodzi od głównego obszaru jego zastosowań, jakim jest sterowanie urządzeniami elektronicznymi, takimi jak: urządzenia biurowe, urządzenia medyczne (w tym implanty), zdalnego sterowania, elektronarzędzia, systemy sterowania silnikami samochodowymi, a nawet zabawki i inne systemy wbudowane.

Mikrokontroler stanowi użyteczny i całkowicie autonomiczny system mikroprocesorowy, nie wymagający użycia dodatkowych elementów, których wymagałby do pracy tradycyjny mikroprocesor. Skądinąd, mikrokontrolery przystosowane są do bezpośredniej współpracy z różnymi urządzeniami zewnętrznymi, w tym również takimi, do których obsługi tradycyjny mikroprocesor wymagałby użycia dodatkowych układów peryferyjnych.

Literatura podstawowa:

1. Rafał Baranowski, „Mikrokontrolery AVR ATmega w praktyce”, BTC, 2005
2. Elliot Williams „Programowanie układów AVR dla praktyków”, Helion, 2014
3. Paweł Borkowski „AVR & ARM7 Programowanie mikrokontrolerów dla każdego”, Helion, 2010
4. arduino.cc Adres strony projektu Arduino
5. microchip.com Adres producenta mikrokontrolera (dawniej Atmel)

Literatura uzupełniająca:

1. Tomasz Francuz „Język C dla mikrokontrolerów AVR”, Helion, 2015
2. D. Guinard, V. Trifa „Internet rzeczy”, Helion, 2017
3. <https://mikrokontroler.pl/>
4. <https://forbot.pl/blog/>
5. miesięcznik „Elektronika dla wszystkich”, Wydawnictwo AVT

Architektury mikroprocesorów typu CISC i RISC

Ze względu na kryterium **typu listy instrukcji**, wyróżnia się dwie architektury mikroprocesorów:

- architektura **RISC** (ang. *reduced instruction set computer*),
- architektura **CISC** (ang. *complex instruction set computer*).

Cechy architektury CISC:

- **duża liczba rozkazów** (instrukcji) (często powyżej 100), występują instrukcje wspierające języki wysokiego poziomu,
- **mała optymalizacja cykli rozkazowych** – niektóre rozkazy potrzebują dużej liczby cykli procesora do wykonania,
- **występowanie złożonych, specjalistycznych rozkazów**, które współpracują na ogół tylko z określonymi rejestrami i
- **wymagają** stosowania określonych trybów adresowania,
- **duża liczba trybów adresowania**,
- do pamięci może się odwoływać bezpośrednio duża liczba rozkazów,
- mniejsza od procesorów RISC częstotliwość taktowania procesora,
- powolne działanie dekodera rozkazów,
- każda instrukcja może wykonać kilka operacji niskiego poziomu, jak na przykład pobranie z pamięci, operację arytmetyczną, albo zapisanie do pamięci a to wszystko w jednej instrukcji.

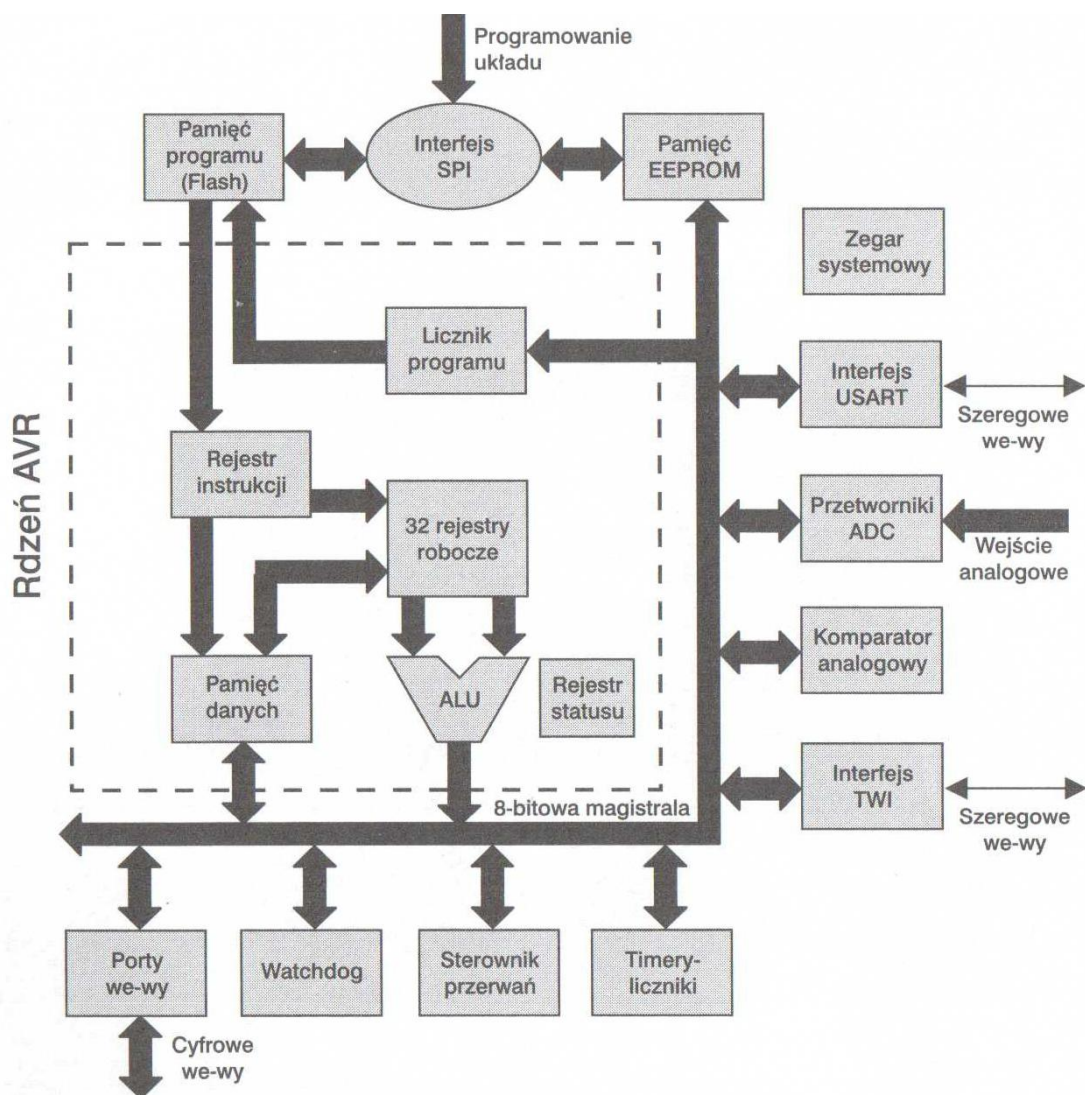
Przykładowe mikroprocesory, mikrokontrolery o architekturze CISC: rodzina x86, 8051

Cechy architektury RISC:

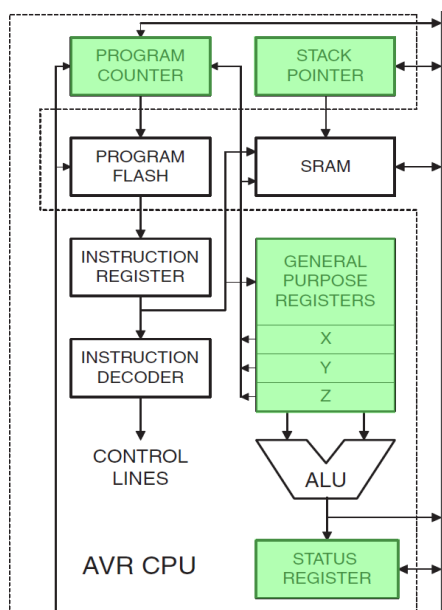
- Zredukowana **liczba rozkazów** do niezbędnego minimum. Ich liczba wynosi kilkadziesiąt, podczas gdy w procesorach CISC sięga setek.
- Zbiór realizowanych instrukcji jest **ograniczony** i spełnia warunki **ortogonalności** (symetrii).
- Zmniejszona **liczba trybów adresowania**, dzięki czemu kody rozkazów są prostsze, bardziej zunifikowane.
- Zwiększenie **liczby rejestrów roboczych** (np. do 32, 192, 256), co ma wpływ na zmniejszenie liczby odwołań do pamięci.
- Ograniczenie **komunikacji pomiędzy pamięcią, a procesorem** – do przesyłania danych pomiędzy pamięcią, a rejestrami służą dedykowane instrukcje, które zwykle nazywają się **load** (załaduj z pamięci), oraz **store** (zapisz do pamięci); pozostałe instrukcje mogą operować wyłącznie na rejestrach (schemat działania na operandach znajdujących się w pamięci jest następujący: załaduj daną z pamięci do rejestru, na zawartości rejestru wykonaj działanie, przepisz wynik z rejestru do pamięci).
- Procesor jest **zbudowany zgodnie z architekturą harwardzką** (w przypadku mikrokontrolerów),
- Procesor wykorzystuje **przetwarzanie potokowe** (*pipeling*) w celu zwiększenia szybkości wykonywania programu, gdyż wszystkie rozkazy wykonują się w jednym cyklu maszynowym, co pozwala na znaczne uproszczenie bloku wykonawczego.

Przykładowe mikroprocesory, mikrokontrolery o architekturze RISC: rodzina AVR, PIC (8-bitowe), ARM (32-bitowe).

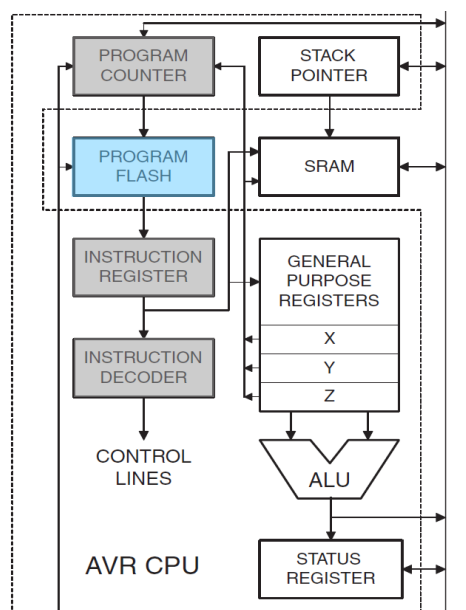
Obecnie popularne procesory firmy Intel, czy AMD z punktu widzenia programisty są widziane jako CISC, ale ich rdzeń jest typu RISC.



Rdzeń mikrokontrolera AVR i typowe układy peryferyjne

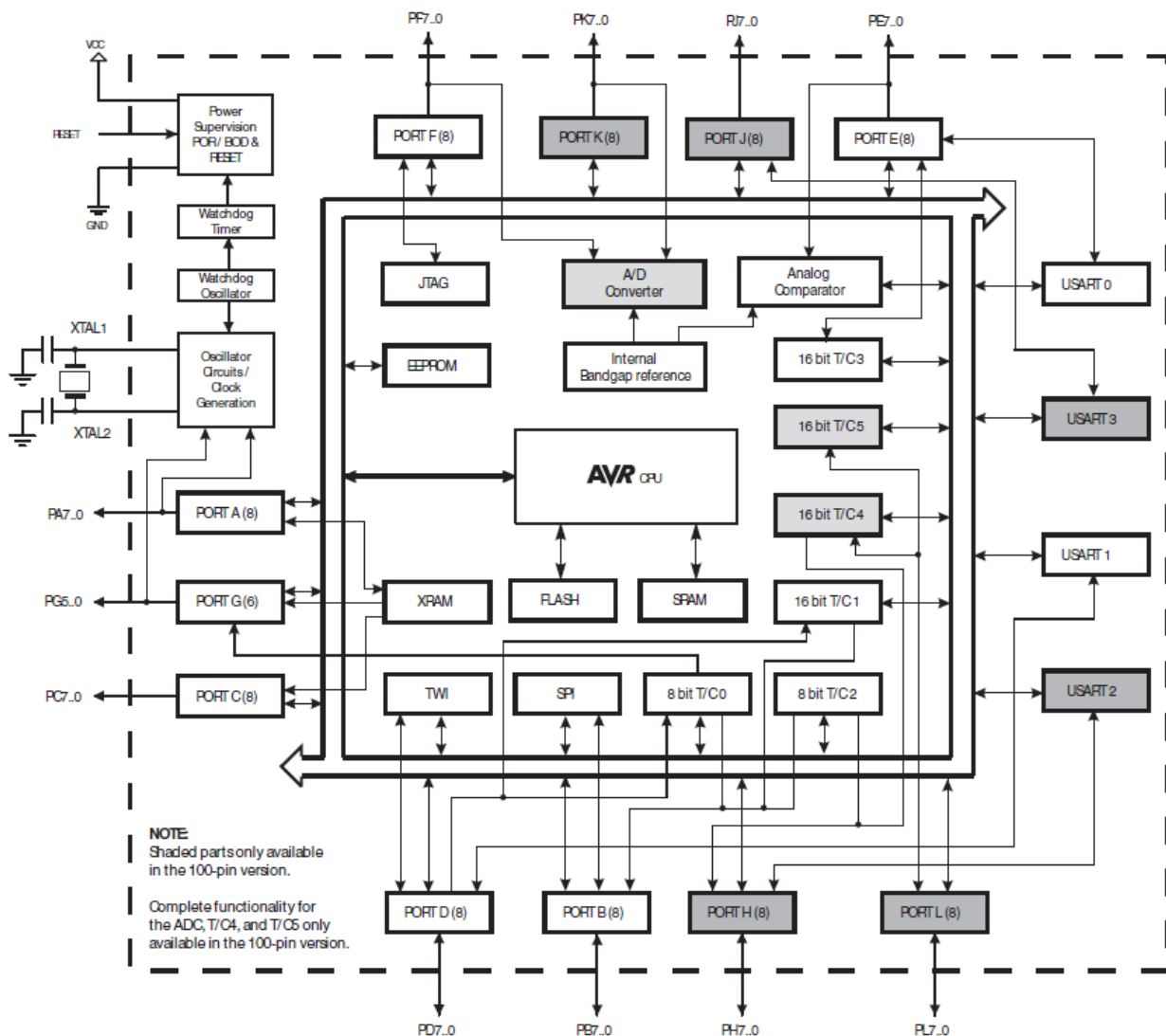


Rejestry procesora AVR dostępne programowo



Rejestry procesora AVR związane z pobraniem i wykonaniem rozkazu

Schemat blokowy procesora ATmega 2560



CECHY

- Wysoka wydajność, niski pobór mocy AVR 8 bitowy mikrokontroler
- Zaawansowana architektura RISC
 - 135 rozkazy – większość wykonywana w czasie jednego cyklu maszynowego
 - 32 x 8 rejestry ogólnie dostępne + rejestr kontroli zewnętrznej
 - statyczne operacje
 - do 16 MIPS przy 16MHz
 - wbudowany dwu-cyklowy układ mnożący
- trwała pamięć programu i danych
 - 256KB wewnętrznej reprogramowalnej Flash o wytrzymałości 10000 cykli zapisu/wymazania
 - opcjonalna sekcja programu ładującego z niezależnymi bitami blokującymi, wewnętrzne programowanie oraz operacje odczytu podczas zapisu

- 4KB EEPROM o wytrzymałości 100 000 cykli zapisu/wymazania
 - 8KB wewnętrznego SRAM
 - do 64KB opcjonalnej zewnętrznej przestrzeni pamięci o blokada programowania dla bezpieczeństwa programów
 - interfejs SPI dla systemowego programowania
- interfejs JTAG
 - zgodności ze standardem JTAG
 - rozległe wbudowane wsparcie debugera
 - programowanie Flash, EEPROM, bezpieczników i bitów blokujących przez interfejs JTAG
- układy peryferyjne
 - dwa 8 bitowe liczniki/stopery z oddzielnymi prescalerami i trybami porównywania
 - cztery rozszerzone 16 bitowe liczniki/stopery z oddzielnymi prescalerami, trybami porównywania i trybami przechwytywania
 - licznik czasu rzeczywistego z oddzielnym oscylatorem
 - cztery 8-bitowe kanały PWM
 - 6/12 kanałów PWM z programowalną rozdzielczością od 2 do 16 bitów
 - wyjściowy modulator porównujący
 - 16 kanałowy 10-bitowy ACD
 - bajtowo zorientowany dwuprzewodowy szeregowy interfejs TWI
 - poczwórny programowalny szeregowy USART
 - interfejs szeregowy SPI
 - programowalny licznik Watchdog'a z wbudowanym oscylatorem
 - wbudowany analogowy komparator
- specjalne cechy mikrokontrolera
 - reset przy włączaniu zasilania, programowalny detektor obniżonego napięcia sieciowego
 - wewnętrzny wzorcowy oscylator RC
 - wewnętrzne i zewnętrzne źródła przerwań
 - sześć trybów obniżonego poboru mocy i uśpienia
 - 54/86 programowalnych linii I/O
- napięcia pracy
 - 1.8V – 5.5V dla ATmega256V
 - 4.5V – 5.5V dla ATmega256
- klasa szybkości
 - 0 – 2 MHz ATmega256V
 - 0 – 16 MHz ATmega256

Budowa mikrokontrolerów AVR opiera się na architekturze harwardzkiej. Jedną z głównych cech charakterystycznych architektury harwardzkiej jest rozdzielenie przestrzeni adresowej pamięci programu i przestrzeni adresowej pamięci danych, co uzyskano poprzez zastosowanie oddzielnych magistrali adresowych.

Dzięki temu możliwe było zastosowanie słowa o różnej szerokości dla pamięci programu i pamięci danych, a także uchronienie się od przypadku, w którym dane mogłyby być interpretowane jako instrukcje.

Mikrokontrolery AVR należą do grupy układów o architekturze RISC (Reduced Instruction Set Computer). Architektura 8086 jest określana nazwą CISC (Complex Instruction Set Computer).

Wykonanie jednego rozkazu CISC wymaga zazwyczaj wykonania wielu operacji, co zwykle trwa kilka taktów zegarowych. Większość rozkazów RISC jest realizowana w jednym takcie zegara, co - pomimo krótszej listy rozkazów - zapewnia szybsze wykonywanie programu.

Wbrew pozorom, programy pisane dla procesorów RISC charakteryzują się większą spójnością, a co za tym idzie mniejszym kodem wynikowym. Cechą wyróżniającą mikrokontrolery AVR jest również to, że zaimplementowano w nich wiele rejestrów wewnętrznych, z których każdy może pełnić funkcję akumulatora podczas wykonywania operacji arytmetycznych i logicznych.

Dzięki temu minimalizuje się liczbę wewnętrznych przesłań, międzyrejestrowych, co korzystnie wpływa na szybkość wykonywania programu i jego wielkość. Projektanci rodziny AVR przewidzieli możliwość wykorzystywania trzech par rejestrów jako rejestrów indeksowych używanych w niektórych trybach adresowania.

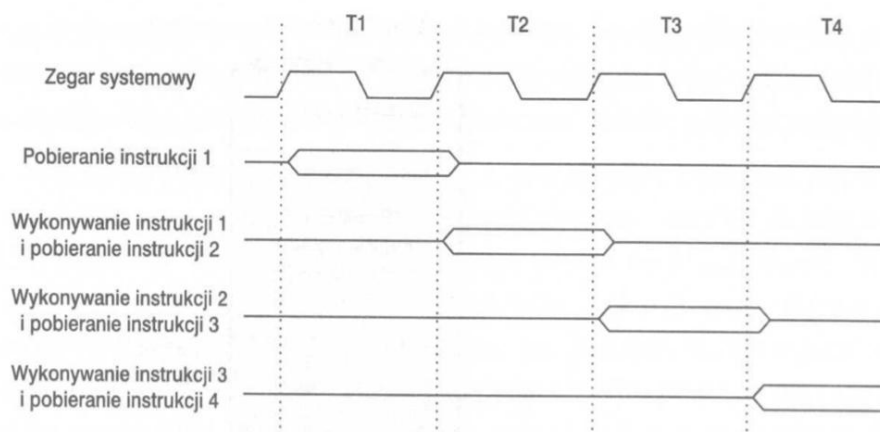
Mikrokontrolery AVR zaliczane do grupy układów 8-bitowych, lecz słowo instrukcji jest 16-bitowe. Taka, a nie inna klasyfikacja wynika z długości rejestrów wewnętrznych i szerokości wewnętrznej szyny danych.

Rdzeń rodziny AVR zaprojektowano z myślą o językach wysokiego poziomu - konkretnie C. Jedną z podstawowych części składowych rdzenia jest jednostka arytmetyczno-logiczna ALU. Może ona wykonywać instrukcje arytmetyczne, logiczne i bitowe, z czego większość jest wykonywana w trakcie jednego cyklu zegarowego. W układach grupy ATmega moduł ten wyposażony jest w układ mnożący, którego praca trwa jedynie dwa takty zegarowe. Co więcej, ALU ma bezpośredni dostęp do wszystkich rejestrów roboczych, z których każdy może być używany jako akumulator.

Większość bloków funkcjonalnych mikrokontrolerów AVR komunikuje się między sobą za pośrednictwem wewnętrznej 8-bitowej magistrali. Dodatkowo prowadzone są pomiędzy nimi niezbędne sygnały sterujące.

Oprócz typowych i na ogół dobrze znanych układów peryferyjnych można wyróżnić bloki, do których użytkownik nie ma bezpośredniego dostępu. Są to: rejestr instrukcji, dekodery instrukcji, wewnętrzny oscylator i układ generujący wewnętrzne sygnały zegarowe. Rejestr instrukcji przechowuje kod aktualnie wykonywanego rozkazu.

Nowoczesność rdzenia AVR przejawia się również w sposobie wykonywania instrukcji. Konstruktorzy zastosowali mechanizm przetwarzania potokowego który polega na jednoczesnym (zachodzącym w jednym cyklu zegarowym) wykonywaniu danej instrukcji i pobieraniu z pamięci kodu instrukcji następnej. Koncepcja ta umożliwia wykonanie w każdym cyklu jednego rozkazu.

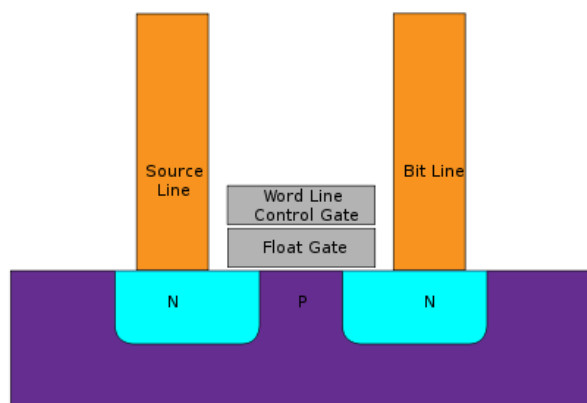


Przetwarzanie potokowe instrukcji

Mimo że mikrokontrolery AVR są układami ośmiobitowymi, ich słowo rozkazowe ma długo szesnastu bitów. Odczyt instrukcji w postaci dwubajtowych słów i wykonywanie ich z zastosowaniem przetwarzania potokowego pozwala osiągnąć znaczne korzyści wydajnościowe.

Większość aktualnie produkowanych mikrokontrolerów dysponuje mechanizmami oszczędzania energii. Podczas normalnej pracy wszystkie bloki mikrokontrolera pracują bez żadnych ograniczeń pobierając ze źródła zasilającego prąd o wartości np. 2,8 mA. W trybie Idle zostaje wstrzymana praca jednostki centralnej (CPU Central Processing Unit), pozostałe bloki (pamięć SRAM, timery/liczniki, port SPI, system przerwań) pracują normalnie. W tym trybie układ pobiera prąd o wartości np. ok. 0,8 mA, a w trybie wyłączenia (Power-down Mode) – 0,1 μ A.

Firma Atmel była jedną z pierwszych, która opanowała technologię wytwarzania pamięci Flash i od samego początku stosowała ją, jako pamięć programu w swoich mikrokontrolerach. Pamięć Flash można wielokrotnie programować, do tego jest możliwe jej programowanie po zamontowaniu mikrokontrolera w systemie. Dzięki temu znacznie uprościły się prace konstrukcyjne i skomplikowane (drogie) programatory stały się zbędne. Do programowania pamięci wystarczy jedynie prosty interfejs.



Budowa komórki pamięci Flash