

Mikrokontrolery

System dydaktyczny z mikrokontrolerem (μ c)
Freescale
(MC68HC908QT4A)



AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

- Mikrokontrolery – wiadomości wstępne
- Mikrokontroler Freescale MC68HC908QT4A
 - cechy charakterystyczne
 - architektura
 - proces wykonywania programu
 - tryby adresacji
 - lista rozkazów
- Koncepcja modułowego stanowiska laboratoryjnego z μC „Nitron” Freescale (d. Freescale Semiconductor)
 - aplikacja – klawiatura i linijka diodowa
 - aplikacja – przetwornik a/c

Uzasadnienie wyboru μ c Freescale

- Bogate wyposażenie w układy peryferyjne, duża ilość dedykowanych wersji
- Dostęp do bezpłatnego firmowego oprogramowania - środowisko uruchomieniowe IDE CodeWarrior (kompilator asemblera, języka C/C++ , debugowanie, graficzny interfejs Processor Expert-dostęp do układów peryferyjnych poza hardwarem)
- Dla studentów: dostęp do bezpłatnych próbek μ c i czujników – współpraca z Freescale System Application Laboratory z siedzibą w Roznowie pod Radhostem, sprawdzona podczas przygotowania dziewięciu prezentacji w ramach Festiwalu Nauki

Mikrokontrolery (μc) nazywane też mikrokomputerami jednoukładowymi (ang. single chip microcomputer) to scalone systemy mikroprocesorowe.

Mikrokontroler (μc) zawiera w jednej obudowie:

- mikroprocesor (μp), określany też jako rdzeń (ang. core)
- pamięć programu (nieulotna: ROM - Read Only Memory, Flash, rzadziej EPROM-Erasable Programmable ROM, EEPROM-Electrically Erasable PROM jako uzupełnienie)
- pamięć danych (RAM - pamięć o dostępie swobodnym)
- układy peryferyjne (interfejsy)

Układami peryferyjnymi mogą być:

- Linie wejścia-wyjścia (binarne), zgrupowane w tzw. porty,
- Porty szeregowo różnych standardów:
 - SCI (Serial Communication Interface – transmisja asynchroniczna),
 - SPI (Serial Peripheral Interface – transmisja synchroniczna),
 - I2C (Inter Integrated Circuit - do komunikacji międzyukładowej),
 - USB (Universal Serial Bus – łączy szeregowo w sprzęcie komputerowym),
 - Ethernet (stosowany w LAN – rozległe sieci komputerowe),
 - CAN (Controller Network Area – magistrała motoryzacyjna),
 - IrDA (Infrared Data Association - standard bezprzewodowej transmisji danych z wykorzystaniem podczerwieni),
- Układy szeregowo-licznikowe: zliczanie impulsów, odmierzanie czasu, generacja przebiegów, wyjścia typu PWM (Pulse Width Modulation) – modulacja szerokości impulsów,
- Przetworniki a/c

Rodzina mikrokontrolerów

Mikrokontrolery zawierające ten sam mikroprocesor tworzą rodzinę.

Cechami wspólnymi rodziny są:

- Ten sam μp czyli rdzeń (ang. core) tu: HC08
- Identyczna lista rozkazów

W ramach rodziny poszczególne typy μc różnią się:

- wewnątrz - zestawem układów peryferyjnych oraz pojemnościami pamięci programu i danych,
- a zewnątrz - ilością pinów obudowy.

Rodzina może zawierać nawet kilkaset układów scalonych.

Rodzina dysponuje wersjami uruchomieniowym, w których jest pamięć programu typu Flash oraz układami docelowymi, stosowanymi przy produkcji wielkoseryjnej z pamięcią programu, określaną przez indywidualną maskę (pamięć ROM).

W ramach rodziny HC08 wyróżniamy 17 grup: AB, AP, AS/AZ, BD, EY, G, GZ, JB/JG/JT, K, LB, LJ/LK, LT, LV, MR, Q, RF, SR .

W grupie Q , jest 29 typów μc : QB, QC, QL, QT, QY, HLC...QT, HLC...QY.

Rodzina mikrokontrolerów

Stosowany w naszym systemie dydaktycznym μ c należy do rodziny 68HC08 firmy Freescale typ QT, o nazwie handlowej „Nitron” (ang. Nitron - nitrogliceryna) oznaczony MC68HC908QT4A (8 pinów). Charakteryzuje się optymalnym stosunkiem ceny do możliwości – duża częstotliwość taktowania, bogate wyposażenie w układy peryferyjne, pamięć Flash, mała obudowa (8 pinów).

μ p komunikuje się z układami peryferyjnymi przy pomocy rejestrów specjalnych (sterującymi, statusowymi i danych), umieszczonych w obszarze pamięci danych (HC08: 64 bajty RAM). Po dodaniu interfejsów, zwiększa się ilość rejestrów, pozostaje taka sama lista rozkazów.

- μ c zawierają specyficzne podzespoły:
- zegar czuwania (watchdog), układ nadzoru zegara, wbudowaną logikę portu testowego. Port taki łączy μ c z zewnętrznym układem nadrzędnym (host) w celu zapisu programu i testowania pracy on-line. Zwykle układ host (typowo komputer) wyposażony jest w oprogramowanie firmowe (np. CodeWarrior).

Po resecie

Sygnał zerowania wewnętrznym sygnałem IRST (internal reset signal), który może być wytworzony przez:

- zewnętrzne wejście sygnału zerowania
- zerowanie po wykryciu włączenia napięcia zasilającego POR (power-on reset)
- układ wykrywający spadek napięcia zasilającego – LVI (low-voltage inhibit)
- zegar czuwania (gdy nastąpiło jego przepełnienie – zegar nie był wyzerowany)
- pobranie nieprawidłowego kodu rozkazu – ILOP (illegal opcode reset)
- próba pobrania rozkazu z niedozwolonego miejsca – ILAD (illegal address reset)
- Po resecie przerwania są zablokowane, interfejsy są nieaktywne, a linie portów ustawione jako wejścia.

Poufność oprogramowania

- Ustawienie 8 bajtów bezpieczeństwa w rejestrach Flash (można z tego zrezygnować ustawiając w czasie programowania opcję – ignore security failure)

Struktura rejestrów

- μ c komunikuje się z układami peryferyjnymi przy pomocy rejestrów specjalnych, umieszczonych w obszarze pamięci danych. Tak więc po dodaniu interfejsów, zwiększamy ilość rejestrów zostawiając taką samą listę rozkazów.
 μ c zawierają specyficzne podzespoły:
- zegar czuwania (watchdog), układ nadzoru zegara, wbudowaną logikę portu testowego. Port taki łączy μ c z zewnętrznym układem nadrzędnym (host) w celu zapisu programu i testowania pracy on-line. Zwykle układ host (typowo komputer) wyposażony jest w oprogramowanie firmowe (np. CodeWarrior).

Struktura rejestrów (c. d.)

Na wyjścia portu PTA mogą być podłączone linie:

- wejścia-wyjścia PTA0÷5;
- timera TCH0,1 ,TCK ;
- przetwornika AD0÷3 ;
- klawiatury KBI0÷5,
- oscylatora zewnętrznego OSC1,2.

O tym która linia będzie podłączona decydują ustawienia w rejestrze/rejestrach sterujących. Czasem jest to kilka rejestrów, czasem jest to wspólny rejestr statusowo - sterujący (np. w przetworniku a/c).

Po resecie wszystkie interfejsy nieaktywne, a linie portu A jako wejścia.

- Wyróżniamy rejestry specjalne:
- sterujące (konfiguracyjne i aktywacyjne)
- statusowe (dają „podgląd” tego co dzieje się w rejestrze)
- danych.

Przerwanie

Są dwa rodzaje przerwań – maskowalne (sprzętowe zewnętrzne – sygnalizowane przez $\overline{IRQ}$ (external interrupt) lub od modułu klawiatury - każda z sześciu linii modułu klawiatury $\overline{KBI}_n = 0 \div 5$) i niemaskowalne (programowe SWI – Software Interrupt).

- Dwa poziomy blokady przerwań.
- Główny bit I = 1 generalnie blokuje przyjmowanie przerwań (po resecie I = 1). Jeśli bit I = 0 to przyjęcie przerwania zależy od bitu zezwalającego w rejestrze sterującym danego interfejsu. Jeśli zablokujemy przerwanie w rejestrze sterującym, a interfejs żąda obsługi przerwania, to stan ten możemy rozpoznać po ustawieniu odpowiedniego bitu w rejestrze statusowym. Możemy pracować w systemie przeglądania (polling) – czytamy rejestry statusowe poszczególnych interfejsów. Przy pollingu można też skorzystać ze zbiorczych rejestrów statusowych, grupujących wyłącznie bity żądań, co przyspiesza przeglądanie.

Tryby oszczędzania energii (niskiego poboru mocy)

Typowe działanie μC w którym jednostka centralna wykonuje kolejne instrukcje programu nazywane jest trybem aktywnym (active mode, run mode) lub normalnym (normal mode). Wartość prądu pobieranego $R_{I_{DD}}$ zależy od:

- temperatury
- realizowanego zadania (liczbą uruchomionych modułów np. ADC, ilością aktywnych we-wy)
- częstotliwość zegara magistrali
- wartość napięcia zasilającego

Dwa tryby oszczędzania energii:

- oczekiwanie (wait mode)
- zatrzymanie (stop mode)

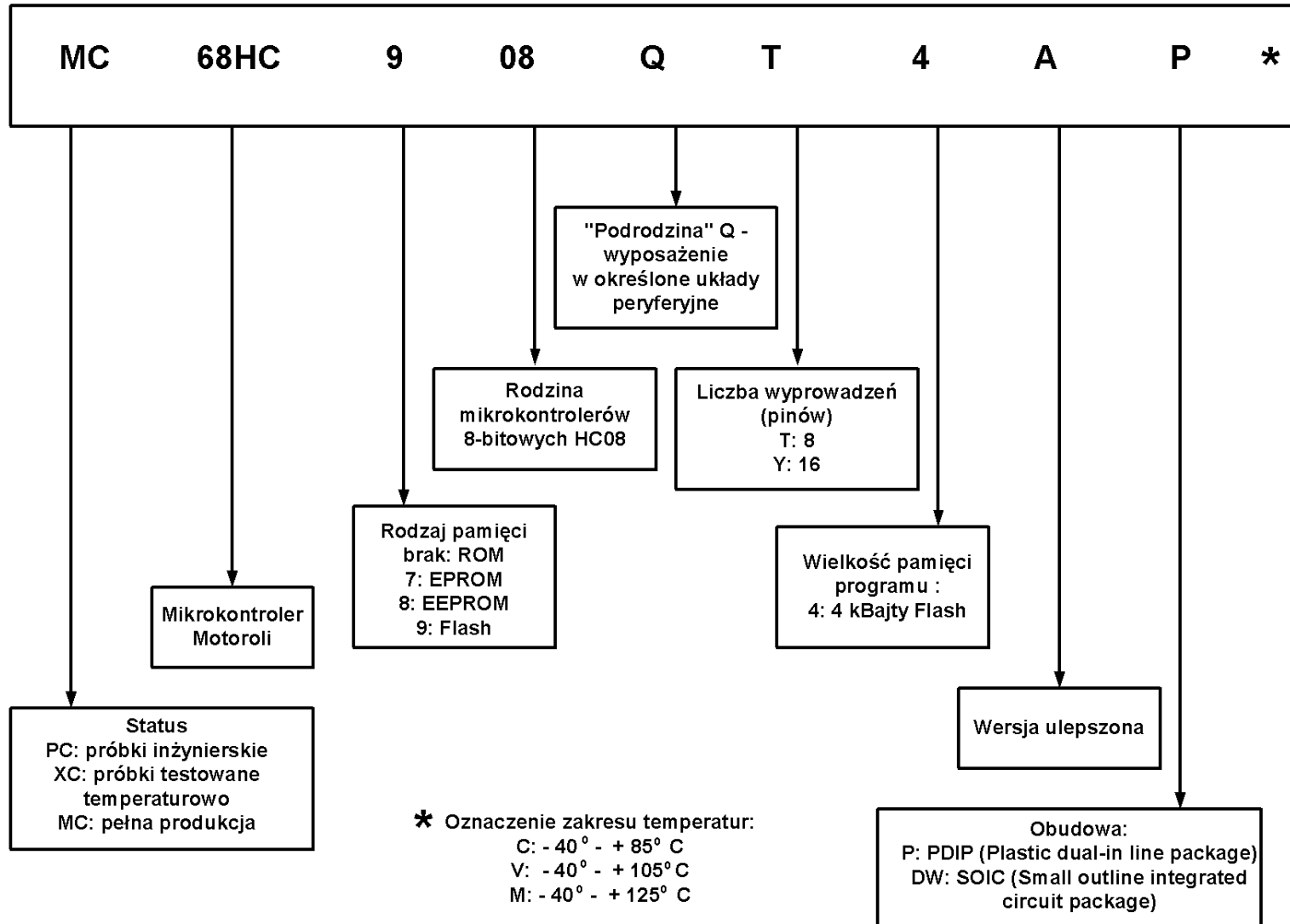
Instrukcje WAIT i STOP wprowadzają μC w stan zawieszenia o niskim poborze mocy.

- Instrukcja WAIT - wejście w tryb oczekiwania, zatrzymanie CPU (zatrzymanie modułu zegara OCS, ustawienie $I=0$ w rejestrze CCR), działają układy peryferyjne i można przerwaniem wyprowadzić CPU z tego trybu przez przerwanie i przez RESET.
- Instrukcja STOP – wejście w tryb zatrzymania, zatrzymanie CPU i układów peryferyjnych, (ustawienie $I = 0$ w rejestrze CCR), jest podtrzymywana pamięć danych – można wyjść z tego trybu tylko przez RESET.

Moduł AWU (Auto Wakeup Module)

- Generuje okresowe przerwania wewnętrzne w czasie trybu STOP, aby wyprowadzić MCU z tego trybu, można wyjść z tego trybu bez sygnałów zewnętrznych. Zezwolenie na uruchomienie zegara w trybie STOP ustawiane jest w rejestrze CONFIG2. Wybierany okres „timeout” w rejestrze CONFIG1. Żądanie wakeup jest traktowane jak przerwanie od klawiatury. Dwa źródła zegara dla AWU: generator RC lub sygnał w module generatora. Te generatory są nieaktywne podczas normalnej pracy. AWU dzieli przerwanie z wektorem KBI.

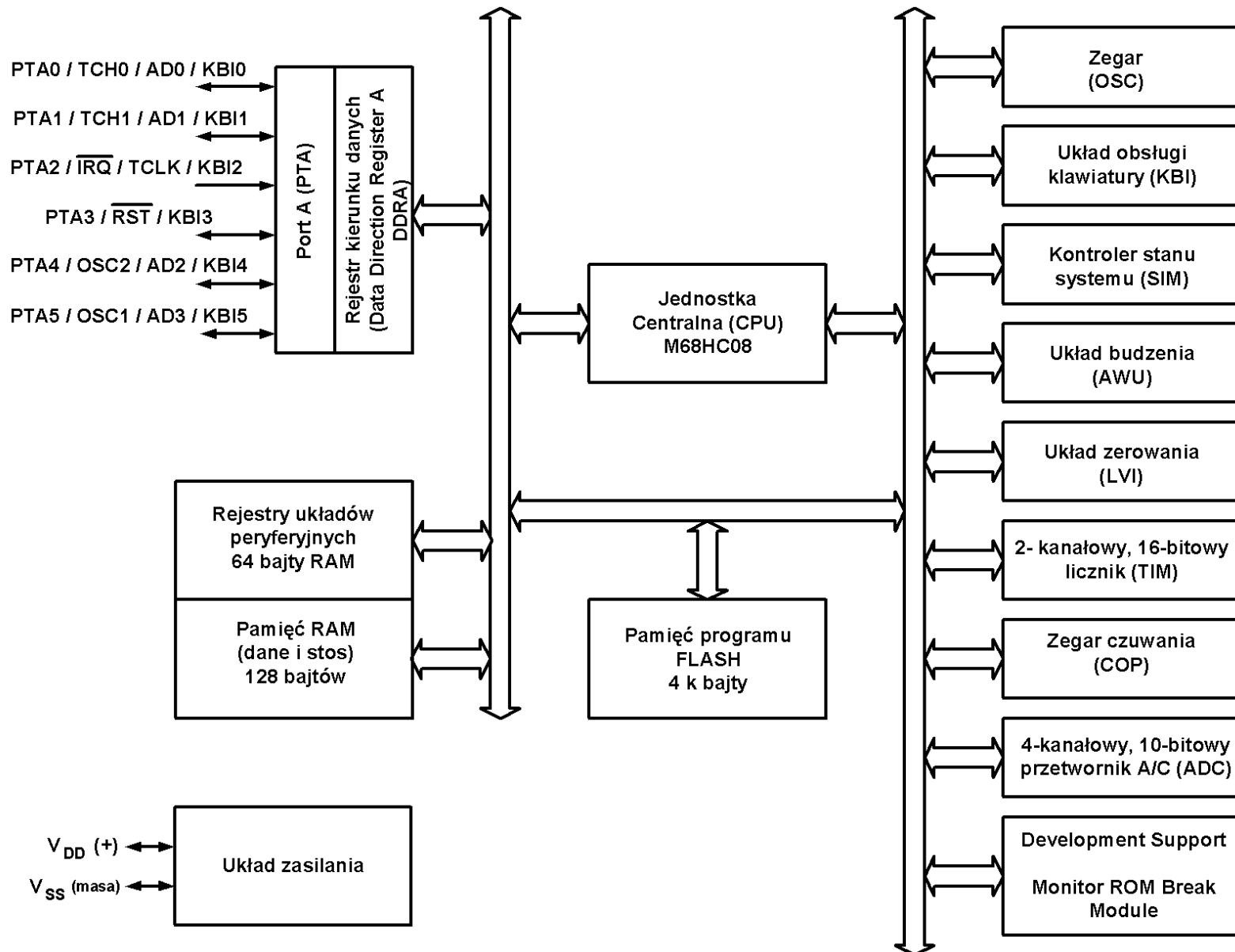
Oznaczenie μ c grupy Q



Cechy μ c MC68HC908QT4A

- Zakres napięć zasilających V_{DD} : 5V i 3V.
- Częstotliwość pracy wewnętrznej magistrali: 1MHz, 2MHz, 3.2MHz (jest to $\frac{1}{4}$ częstotliwości wewnętrznego zegara)
- Wewnętrzna pamięć programu Flash 4 k Bajtów
- Port równoległy PTA
- Podzespoły przerwań zewnętrznych: - IRQ, - KBI
- 2-kanałowy, 16-bitowy moduł interface'u timera
- 2-kanałowy, 16-bitowy przetwornik a/c
- Dwa tryby oszczędzania energii: Wait i Stop
- Zegar czuwania COP
- Tryb pracy MONITOR (program w pamięci ROM μ c) – wspomaganie uruchamiania (development support) czyli programowanie pamięci Flash przez RS232 lub USB z komputera zewnętrznego (host).

Schemat blokowy mikrokontrolera MC68HC908QT4A



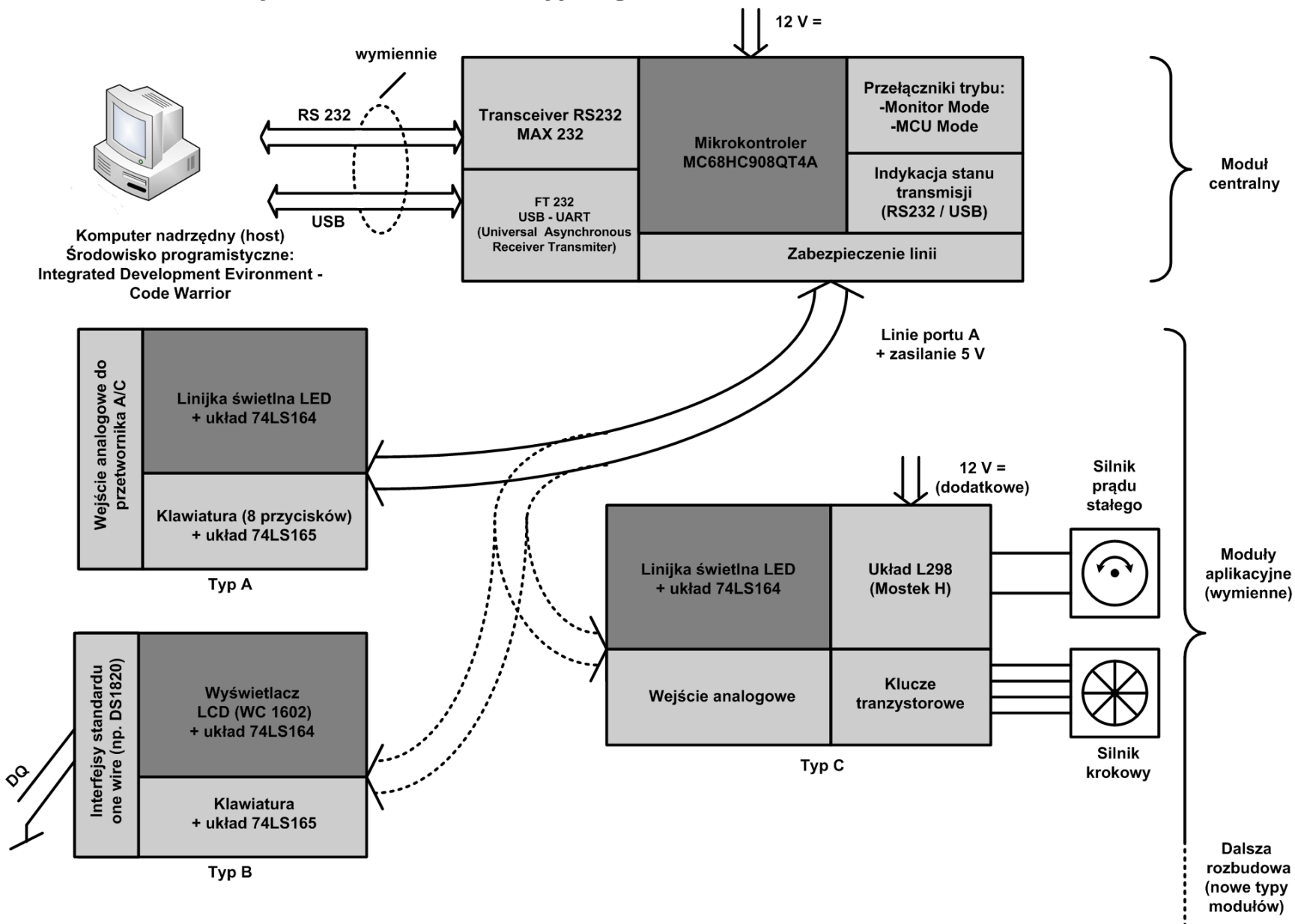
Opis modułów – user mode

- Moduł OSC – wewnętrzny generator: $f = 4$ lub 8 lub $12,8$ MHz . $12,8$ MHz jest domyślną częstotliwością po resecie. Zegar magistrali ($1/4$ częstotliwości wew. generatora) : 1 lub 2 lub $3,2$ MHz. Możliwość trymowania.
- Moduł COP (Computer Operating Properly) czyli zegar czuwania – zabezpieczenie przed uszkodzeniem pamięci flash lub błędnym programem. Zawiera licznik, który generuje sygnał zerowania μc po przepełnieniu. Prawidłowy program zeruje stan licznika co pewien czas.
- Moduł LVI (Low Voltage Inhibit) - wykrywający spadek napięcia zasilającego
- Moduł AWU (Auto Wake Up) – generuje okresowe przerwania w celu wyprowadzenia μc ze stanu zatrzymania „Stop” (bez sygnałów zewnętrznych)
- Moduł KBI (Keyboard Interrupt) – generuje przerwania wyzwalane zboczem
- Moduł SIM (System Integration) – wspomaganie przerwań wewnętrznych i zewnętrznych, wspólnie z CPU steruje czynnościami μc
- Moduł przetwornika a/c – 4 kanałowy, 10 lub 8-bitowy, metoda liniowej kompensacji wagowej, czas przetwarzania kilkanaście μs
- Moduł timera – przebiegi czasowe z funkcją Input Capture , Output Compare, PWM (Pulse Width Modulation)

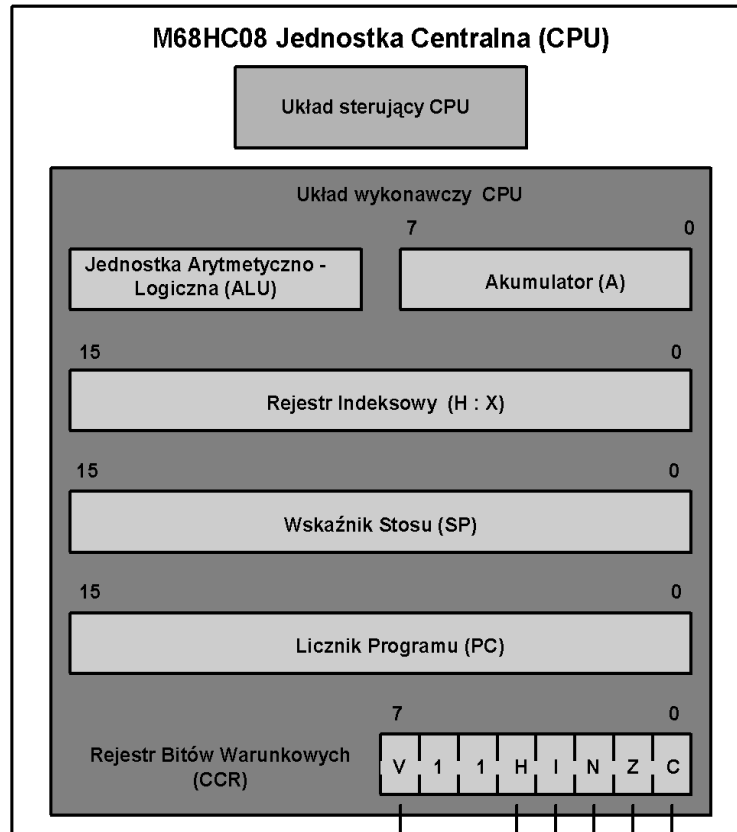
Opis modułu – Monitor mode

- Tryb pracy MONITOR (program w pamięci ROM μ c) – wspomaganie uruchamiania (development support) czyli programowanie pamięci Flash przez RS232 lub USB z komputera zewnętrznego (host).
- Wykorzystanie Modułu Development Support – Monitor ROM Break Module
- Nadrzędny host przejmuje kontrolę nad μ c
- Realizacja programu („firmware”) z pamięci ROM
- Dostęp do stanu całego układu (rejstry peryferyjne, lokacje RAM, pamięć Flash – możliwość poufności oprogramowania na 8 bajtach bezpieczeństwa)
- Możliwość modyfikowania każdego rejestru CPU oraz dowolnej lokacji RAM
- Pułapka sprzętowa (breakpoint) – praca krokowa

Struktura systemu laboratoryjnego z mikrokontrolerem MC68HC 908QT4A



Jednostka centralna



Objaśnienia:

Znacznik liczby ujemnej (kopia najstarszego bitu argumentu: znak w kodzie U2; w kodzie NB informacja, że argument jest większy od połowy)

Motorola - Bity warunkowe statyczne: zerowość i znak - mogą być określone bez działania
np. rozkaz przesłania ustawia bity statyczne przesłanego argumentu

Bity dynamiczne: C, V są wynikiem działania na argumentach

Znacznik przeniesienia połówkowego H i carry C ustawiają się wprost przy dodawaniu i według negacji wartości przeniesionej przy odejmowaniu

Znacznik Przeniesienia / Pożyczki (Carry / Borrow)

Znacznik Zerowości (Zero)

Znacznik liczby ujemnej (Negative)

Maska przerwań (Interrupt Mask)

Znacznik przeniesienia połówkowego (Half Carry)

Znacznik przepełnienia (Overflow)- ma znaczenie tylko dla kodu U2

Moduł oscylatora OSC Module

- Zegar wewnętrzny o małej dokładności (, może być też zewnętrzny: RC, Generator lub kwarc (potrzebne 2 piny).
- Częstotliwość zegara wynosi $\frac{1}{4}$ każdej z poniższego źródła zegara.
- Wewnętrzny oscylator: Generowana wewnętrznie, stała częstotliwość zegara, przestrajana 0,4 % . Można wybrać trzy częstotliwości wewnętrznego oscylatora: 12,8 MHz, 8 MHz, i 4 MHz. 12,8 MHz jest domyślną częstotliwością po resecie. Po podzieleniu przez 4 daje zegar magistrali: 3,2 MHz.
- Wady: nie ma możliwości wyboru odpowiedniej prędkości pracy oraz mała dokładność: nawet do ± 25 %.
- Zewnętrzny oscylator : zewnętrzny zegar może być dołączony tylko do pinu OSC1.
- Zewnętrzny RC: wbudowany moduł oscylatora (oscylator RC) wymaga tylko zewnętrznego R do linii OSC1. Kondensator C jest wewnątrz. Też ma małą dokładność.
- Zewnętrzny rezonator kwarcowy (XTAL oscillator): wbudowany oscylator wymaga zewnętrznego kwarcu – podłączonego do dwóch pinów OSC1 i OSC2. Można wybrać częstotliwość zegara magistrali i uzyskać dużą dokładność.

Moduł oscylatora OSC Module (c. d.)

- Do odmierzania czasu rzeczywistego (RTC – Real Time Clock) musimy odmierzać dokładnie 1 sekundę. Odpowiednio dobrana częstotliwość kwarcu, podzielona w wewnętrznym bloku licznikowym musi zapewniać generację całkowitej ilości przerwań na sekundę (przepełnienie licznika).

Mapa pamięci mikrokontrolera

PAMIĘĆ μ c MC68HC908QT4A

- Pamięć ma jednolitą przestrzeń adresową 64 k Bajtów (adresy \$0000÷\$FFFF), co oznacza, że poszczególne bloki pamięci oraz porty i układy peryferyjne współdzielą jeden zbiór adresów. Te same rozkazy i tryby adresowania mogą być użyte w odniesieniu do lokacji pamięci oraz do dowolnych rejestrów.
- Strona zerowa (page 0) obejmuje 256 lokacji pamięci statycznej RAM (Random Access Memory) o adresach \$0000÷\$00FF. Znajdują się na niej rejestry układów peryferyjnych i stos. Do tej strony stosuje się tryb adresacji bezpośredniej.

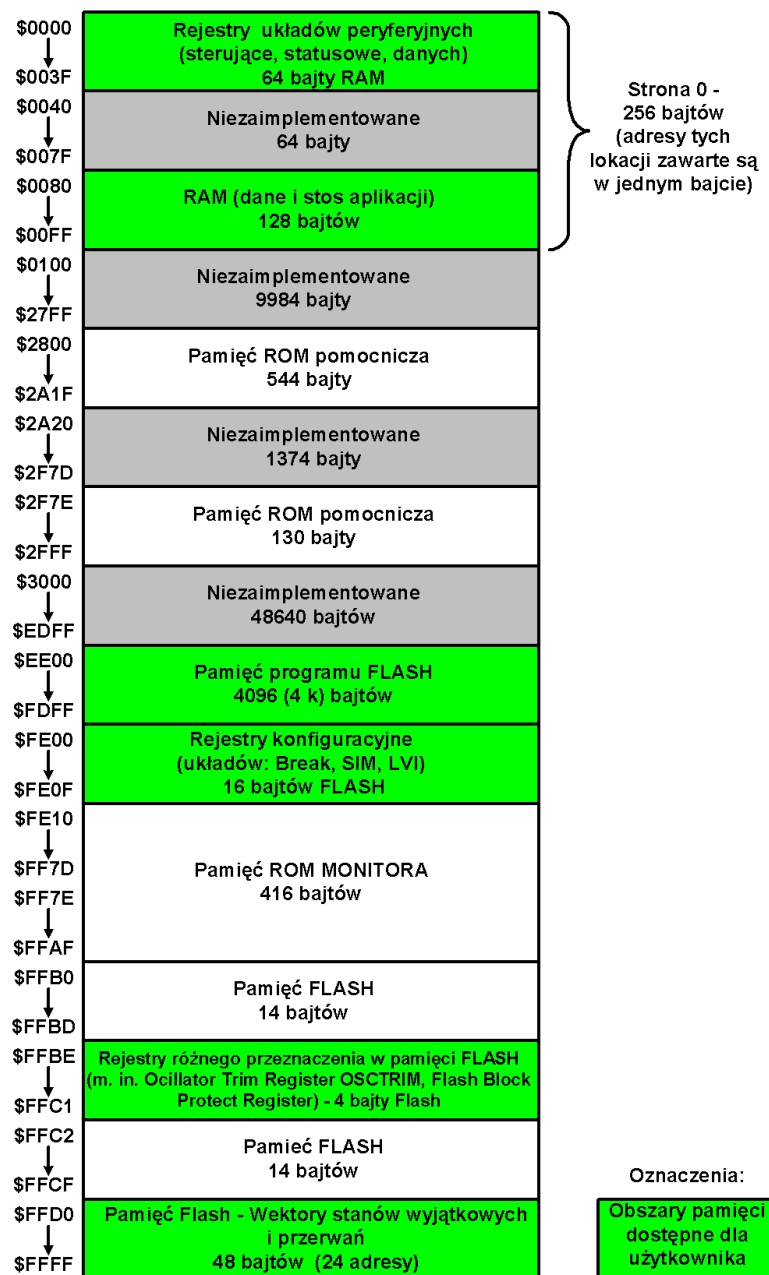


Tabela wektorów

czyli tabela adresów początkowych stanów wyjątkowych i przerw

Priorytet wektora		Moduł lub instrukcja	Adres w Tabeli	Bit aktywujący (flaga)	Bit maskujący	Rejestr
Najniższy	8	Przetwornik A/C - zakończona konwersja	\$FFDE:\$FFDF	(COCO)	AIEN	ADSCR
	7	Moduł obsługi klawiatury - KBI	\$FFE0:\$FFE1	KEYF	IMASKK	KBSCR
	6	Moduł TIM - przepełnienie licznika	\$FFF2:\$FFF3	TOF	TOIE	TSC
	5	Moduł TIM - kanał 1	\$FFF4:\$FFF5	CH1F	CH1IE	TSC1
	4	Moduł TIM - kanał 0	\$FFF6:\$FFF7	CH0F	CH0IE	TSC0
	3	-				
	2	Przerwanie zewnętrzne - $\overline{\text{IRQ}}$	\$FFFA:\$FFFB	IRQF1	IMASK1	INTSCR
	1	Instrukcja SWI	\$FFFC:\$FFFD	-	-	-
	0	Zerowanie - Reset	\$FFFE:\$FFFF	-	-	-
Najwyższy						

Pamięć

- Lokacje pamięci poniżej adresu \$00FF mogą być dostępne przy użyciu bardziej wydajnego trybu adresacji bezpośredniej i każdy pojedynczy bit z tego obszaru może być dostępny przez instrukcje manipulacji bitami (BCLR, BSET, BRCLR i BRSET).
- Pamięć RAM zachowuje dane gdy μ C jest w trybie oczekiwania o niskim poborze mocy (low-power wait) lub w trybie zatrzymania (stop).
- Pamięć FLASH jest przeznaczona do przechowywania programu. Program jest ładowany do pamięci FLASH po końcowej asemblacji.

Wykonanie programu

- W mikrokontrolerach rozkazy wykonywane są sekwencyjnie (w odróżnieniu od przetwarzania potokowego - pipelining, stosowanego w niektórych mikroprocesorach).

Każda faza wykonywania rozkazu realizowane jest po zakończeniu poprzedniej (w potokowym fazy zachodzą na siebie).

Te fazy to:

- pobranie (fetch)
- dekodowanie (decode)
- wykonanie (execute)
- zapis (write)

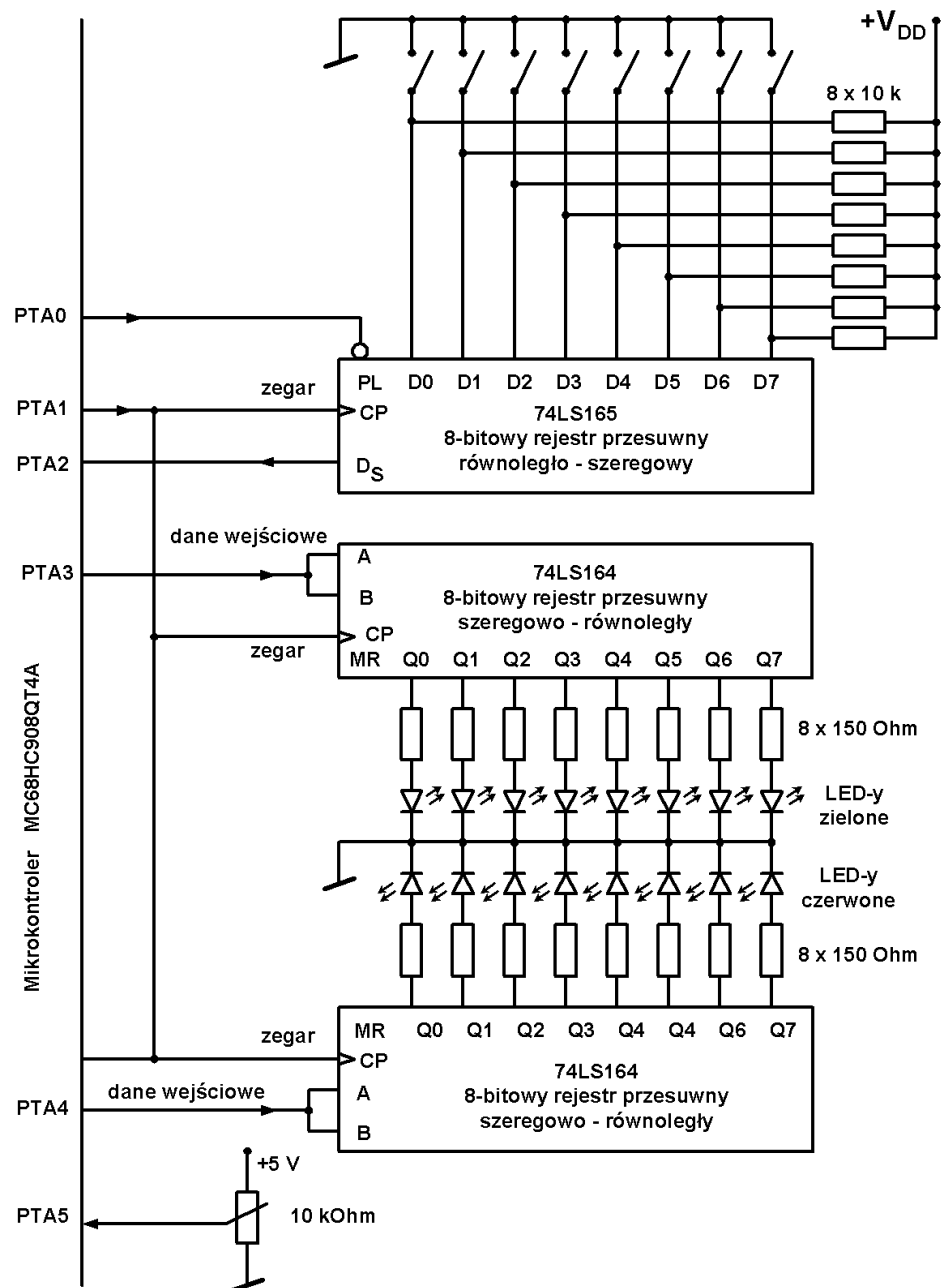
Wykonanie programu

- Mikrokontroler rozpoczyna pracę od stanu początkowego inicjowanego przez sygnał RESET. Jest to stan wyjątkowy (exception). Wtedy rejestry ustawiane są w stan początkowy, a program jest odczytywany z pamięci FLASH począwszy od lokacji \$FFFF (jest to koniec dostępnego 64 k bajtowego obszaru – znajduje się tam tablica wektorów dla stanów wyjątkowych i przerwań). Obsługa stanu wyjątkowego RESET powoduje, że do licznika rozkazów PC wpisywana jest kolejno wartość \$FFFE, adres jest wysyłany na szynę adresową i pobierany jest pierwszy bajt (STARSZY) adresu startowego programu, potem kolejno adres \$FFFF i pobierany jest drugi bajt (MŁODSZY) adresu. Te dwa bajty tworzą adres startowy tzw. wektor startowy. Jest on przygotowany przez programistę w pamięci Flash lub pobrany z bloku ROM jeżeli μC uruchomiono w trybie MONITORA
- Zerowanie powoduje ustawienie niektórych rejestrów: wskaźnik stosu SP - \$00FF, Bajt H w rejestrze indeksowym (H:X) stan \$00, maska przerwań I = 1 w rejestrze CCR (przerwania zablokowane).

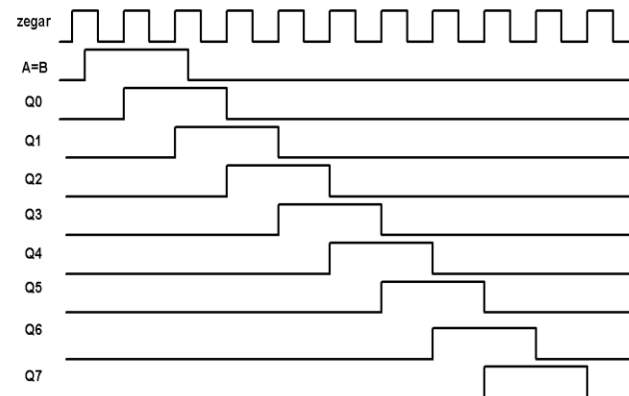
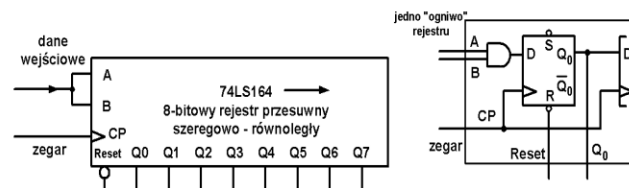
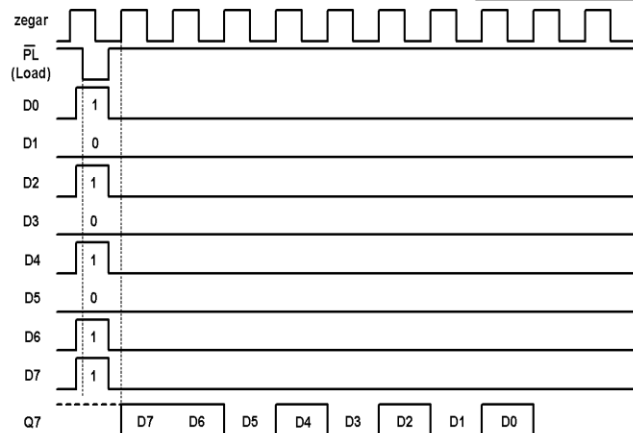
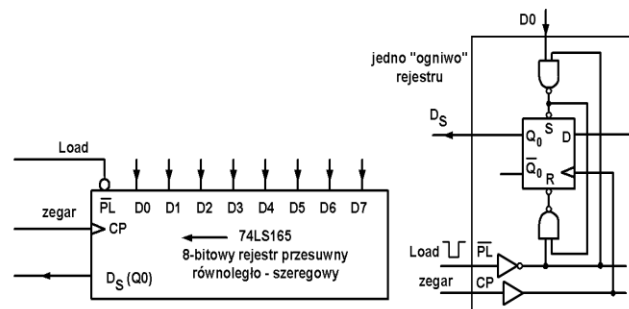
Aplikacja – klawiatura i linijka diodowa

- Przykład zwiększania ilości linii portu A:
 - wejścia (8 przycisków)
 - i wyjścia (16 diod LED)
- przy pomocy rejestrów przesuwnych:
- rejestru równoległo-szeregowego 74LS165
 - i rejestru szeregowo-równoległego 74LS164

Schemat ideowy aplikacji: klawiatura – linijka diodowa

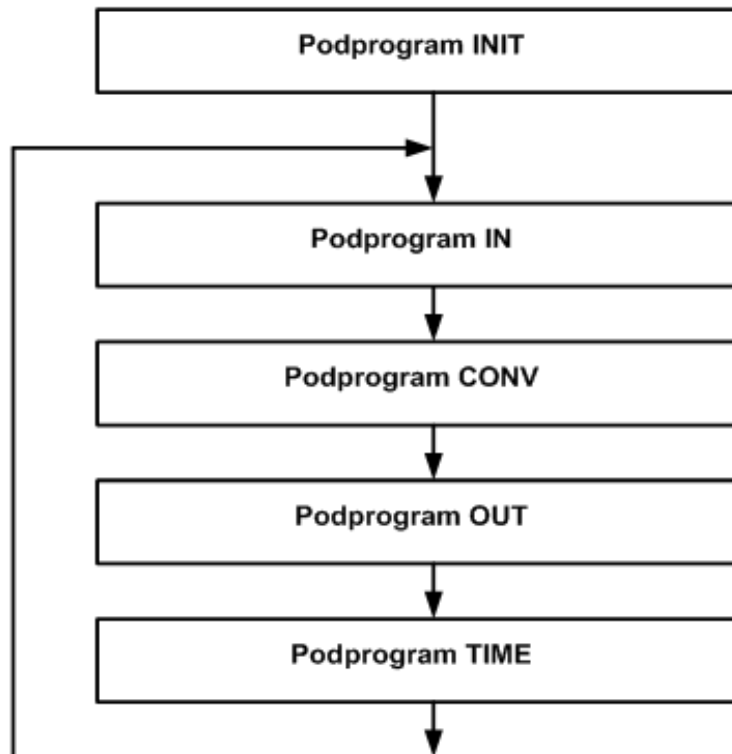


Rejestry przesuwne



Program główny:

- podprogram INIT (przygotowanie mikrokontrolera do pracy)
 - podprogram IN (odczyt stanu klawiatury),
 - podprogram CONV (konwersja odczytu),
 - podprogram OUT (wyświetlenie wyniku),
 - podprogram TIME (opóźnienie programowe),
- skok bezwarunkowy do początku pętli poza czynności inicjalizujące.



Mikrokontrolery

Dziękuję za uwagę



AGH

AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE