

Systemy mikroprocesorowe

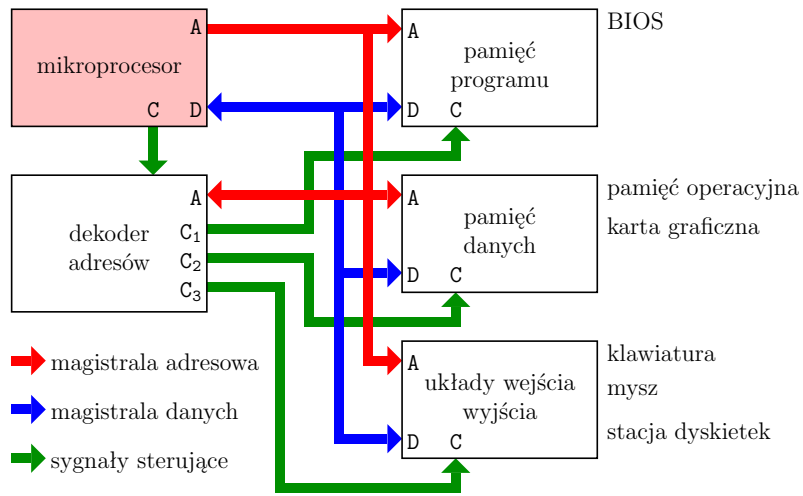
Dariusz Chaberski

Grudziądz, 3 października 2009

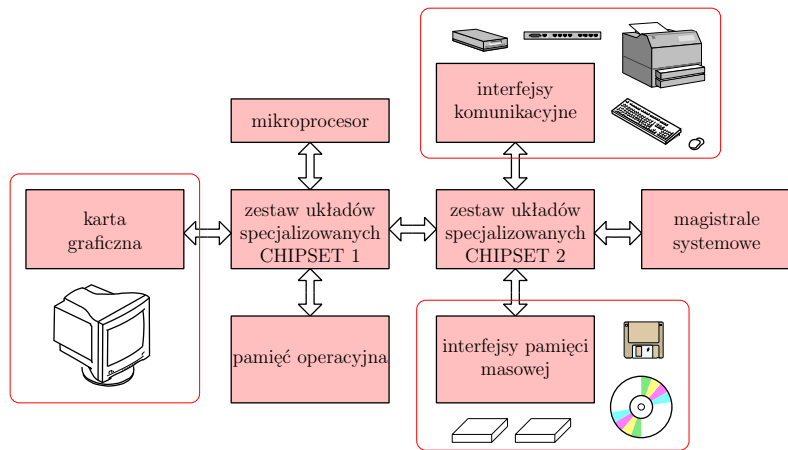
1 Ustalenia

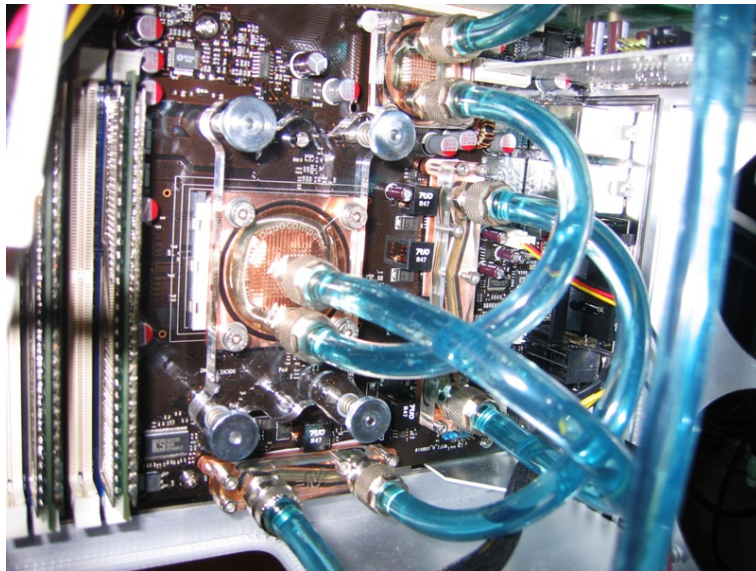
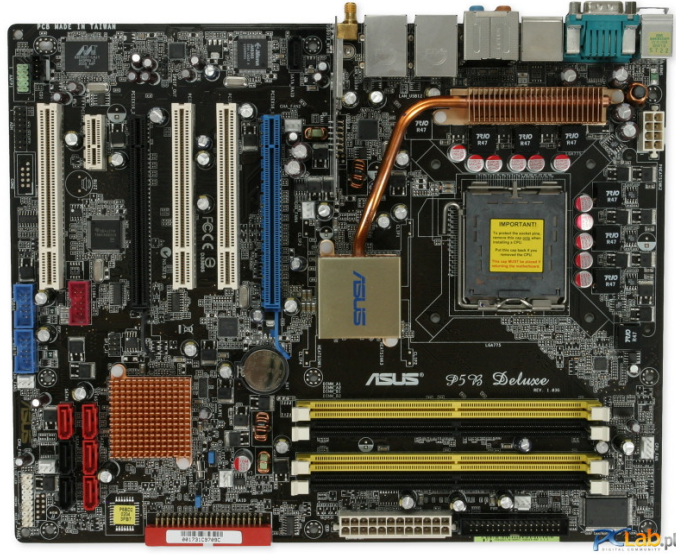
- email:
rkf.chaberski@fizyka.umk.pl
- www:
<http://www.fizyka.umk.pl/~rkf.chaberski>
<http://www.fizyka.umk.pl/rkf/RKFGrudziadz.html>
- 20 godzin = 2 godziny + 6 × 3 godziny
 - Październik 3, 10, 17 24, 31
 - Listopad 7, 14, 21, 28
 - Grudzień 5, 12, 19, 26
 - Styczeń 2, 9, 16

2 Architektura

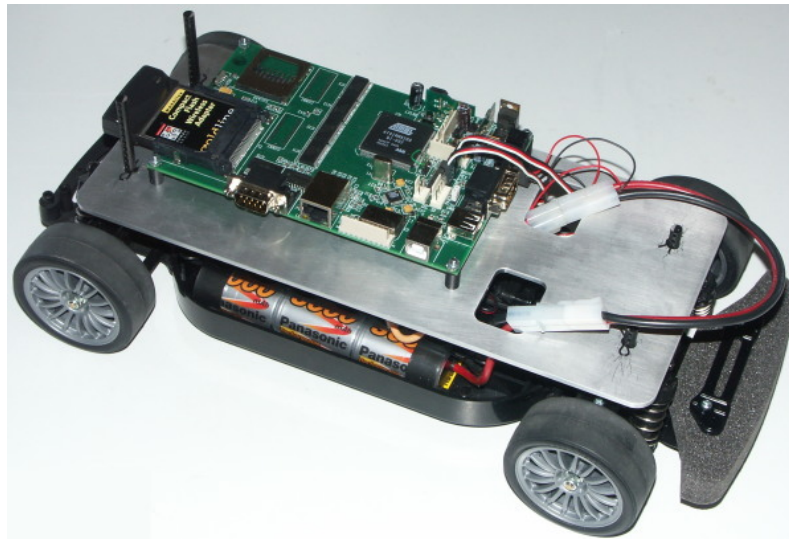
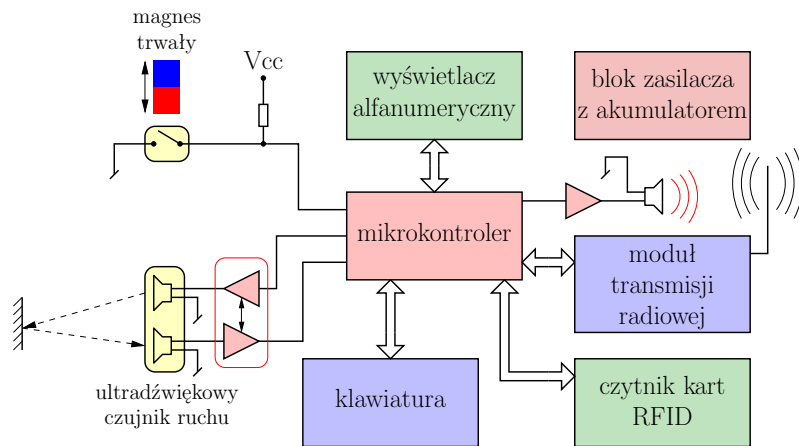


2.1 Komputer osobisty

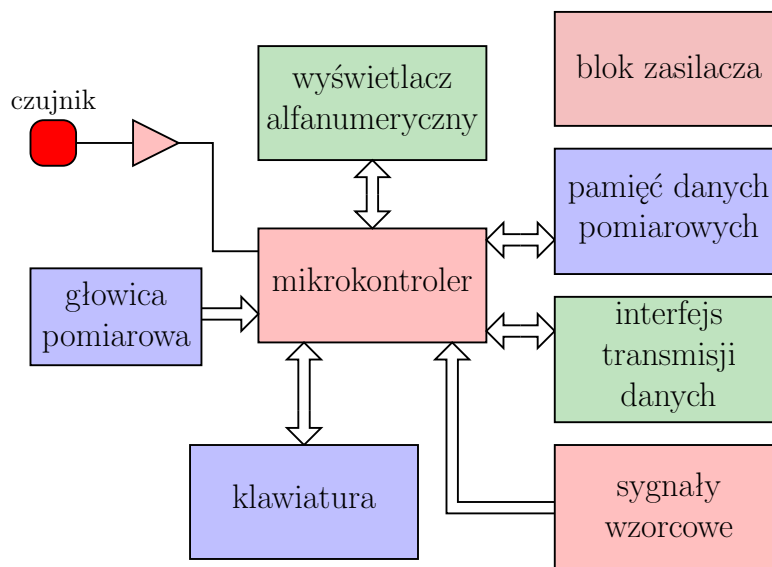




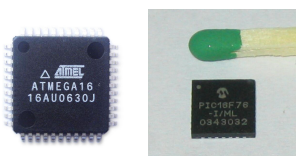
2.2 System kontrolny



2.3 System pomiarowy



2.4 Mikrokontroler - mikroprocesor porównanie



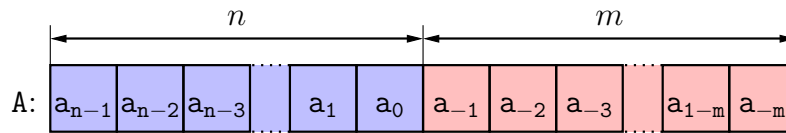
- mikrokontroler
 - mała moc obliczeniowa (30 MIPSów)
 - na ogół brak instrukcji zmiennoprzecinkowych
 - przetwarzanie jednopotokowe
 - bogate peryferia (liczniki i układy czasowe, przetworniki A/C i C/A, duża liczba interfejsów)
 - dostępne operacje bitowe
 - brak układu zarządzania pamięcią MMU
 - praca tylko w trybie rzeczywistym
 - rzadko obsługiwany bezpośredni dostęp do pamięci DMA
 - zastosowanie: aparatura kontrolno pomiarowa



- mikroprocesor
 - duża moc obliczeniowa (10000 MIPSów obecnie)
 - wielordzeniowość oraz technologia HT
 - przetwarzanie wielopotokowe
 - dostępne instrukcje zmiennoprzecinkowe
 - do działania wymaga dodatkowych peryferii (kontroler DMA, pamięć (programu i danych), kontroler przerw)
 - dostępny układ zarządzania pamięcią MMU
 - adresowanie w trybie rzeczywistym i wirtualnym
 - z reguły architektura Von-Neumana
 - możliwość pracy w systemie wieloprocesorowym
 - zastosowanie: komputery (stacje robocze - komputery osobiste, serwery)

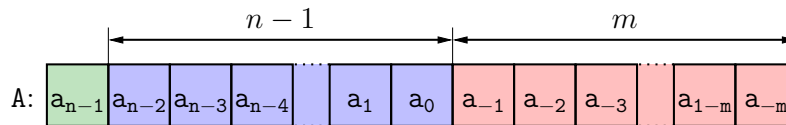
3 Dwójkowe kody liczbowe

3.1 Kod naturalny



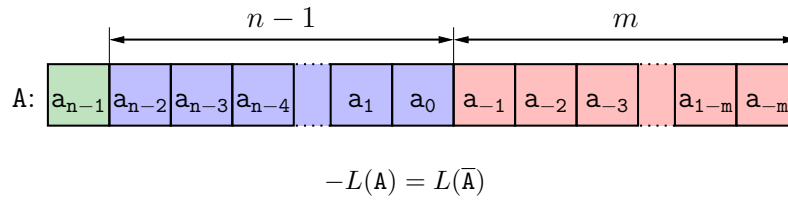
$$L(A) = \sum_{i=-m}^{n-1} 2^i L(a_i)$$

3.2 Kod znak-moduł (ZM)

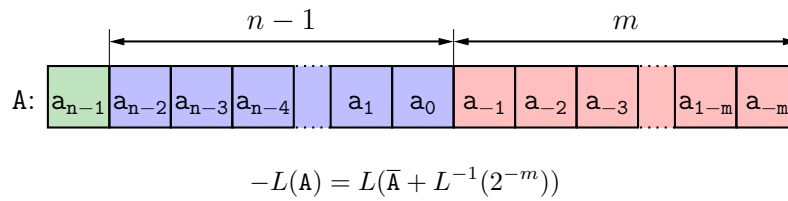


$$L(A) = (-1)^{L(a_{n-1})} \sum_{i=-m}^{n-2} 2^i L(a_i)$$

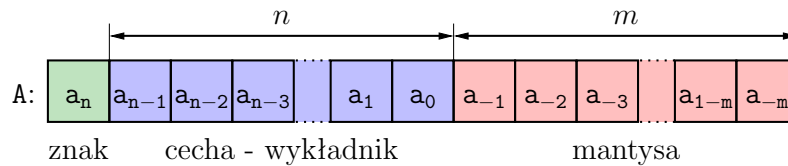
3.3 Kod uzupełnienie do 1 (U1)



3.4 Kod uzupełnienie do 2 (U2)



3.5 Zapis zmiennoprzecinkowy



$$C = A[n - 1, 0]$$

$$M = A[-1, -m]$$

$$S = A[n]$$

$$E = 2^{n-1} - 1$$

$$L(A) = (-1)^{L(S)} L(M) \times 2^{L(C) - E}$$

$$2^n - 1 \geq L(C) \geq 0$$

$$1 \leq L(M) < 2, \text{ normalizacja}$$

$$L(S) = 0, 1$$

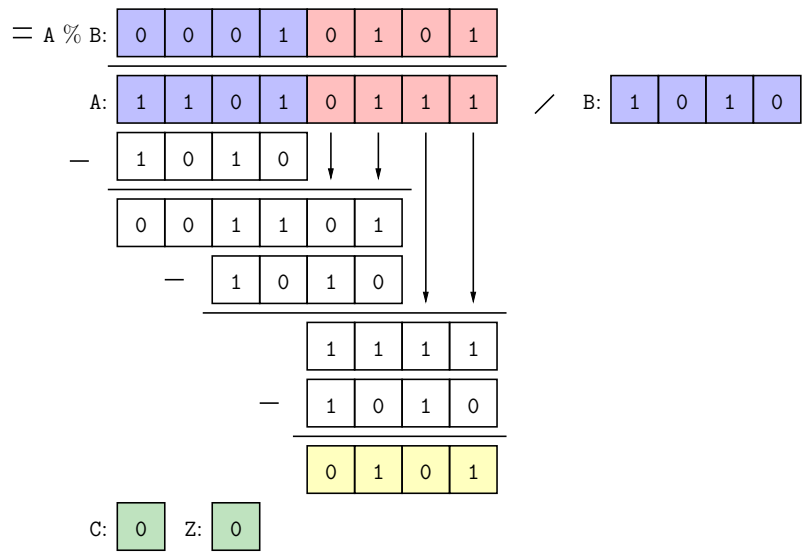
4 Operacje na liczbach dwójkowych

4.1 Suma

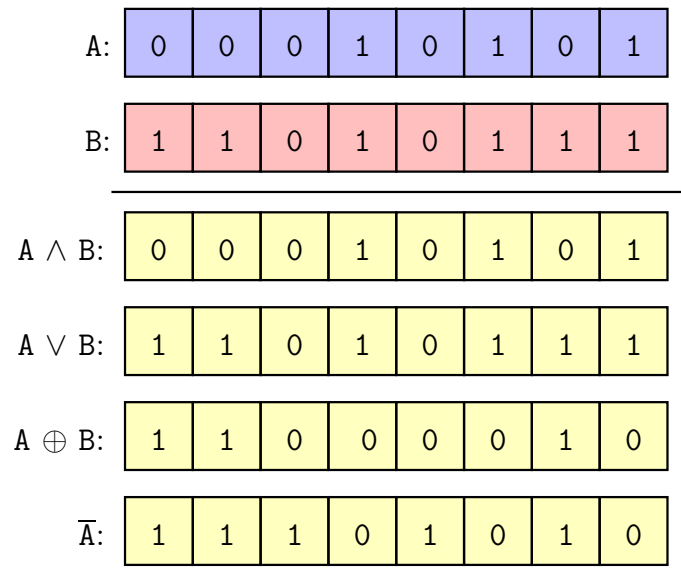
$$\begin{array}{r} \text{A: } 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \\ + \text{B: } 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \\ \hline = \text{A+B: } 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \\ \text{C: } 1 \quad \text{Z: } 0 \quad \text{N: } 0 \quad \text{V: } 0 \quad \text{S: } 0 \quad \text{H: } 0 \end{array}$$

4.2 Różnica

$$\begin{array}{r} \text{A: } 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \\ - \text{B: } 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \\ \hline = \text{A-B: } 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \\ \text{C: } 1 \quad \text{Z: } 0 \quad \text{N: } 1 \quad \text{V: } 0 \quad \text{S: } 1 \quad \text{H: } 1 \end{array}$$

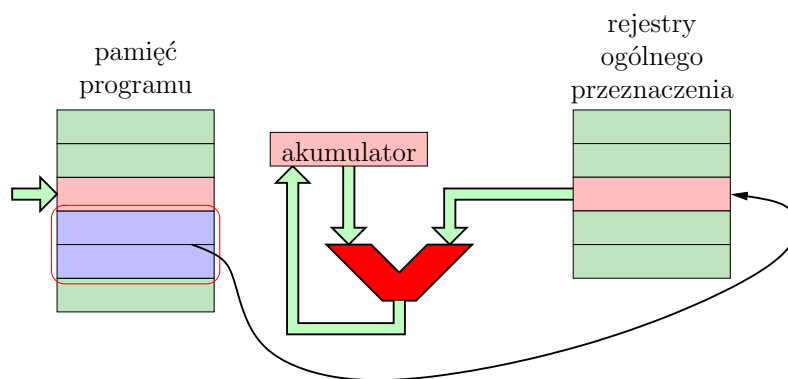


4.5 Operacje logiczne

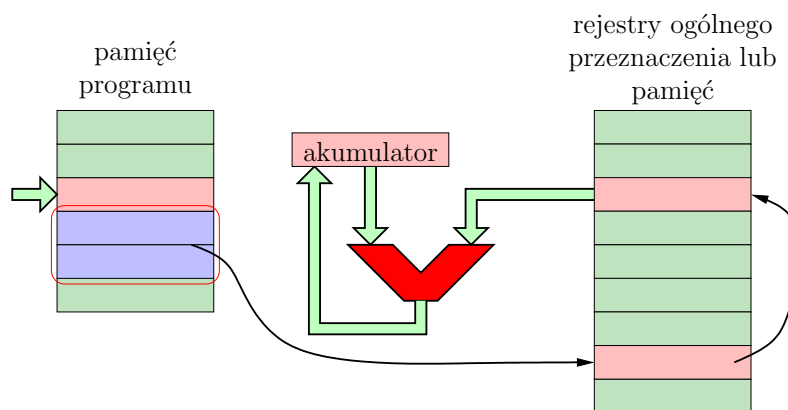


5 Adresowanie

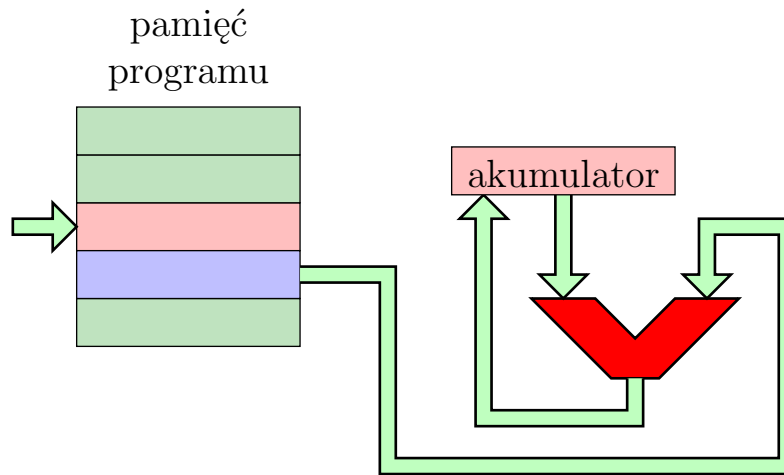
5.1 Adresowanie bezpośrednie



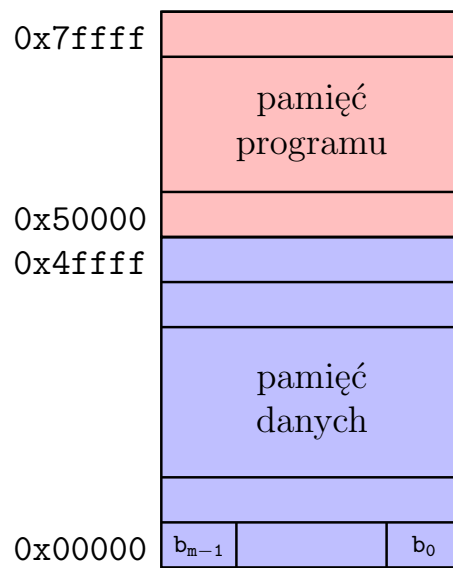
5.2 Adresowanie pośrednie



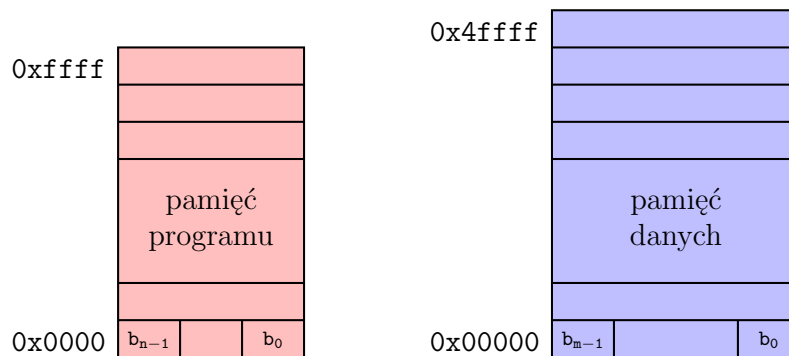
5.3 Wartość natychmiastowa



5.4 Architektura Von Neumana

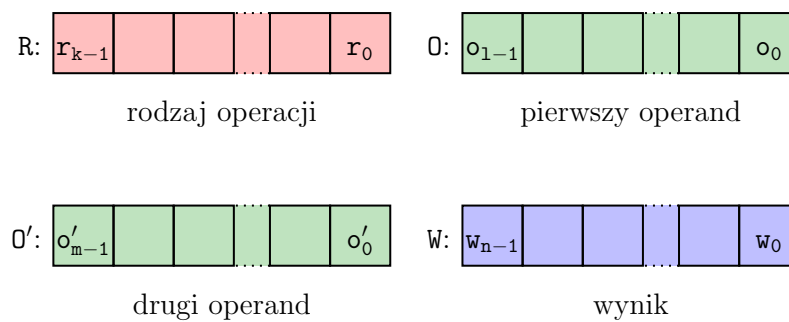


5.5 Architektura harwardzka



6 Instrukcje

6.1 Budowa



$$R(W, F) = R(O, O', F)$$

- przykłady

$$F \circ [W] = [O] \dagger O'$$

$$F \circ \bar{W} = O \dagger [\#O']$$

6.2 Rodzaje instrukcji

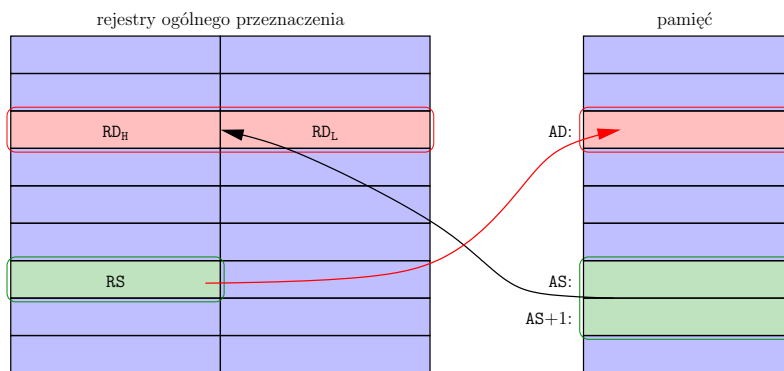
- pełniona funkcja
 - przesłań
 - arytmetyczne
 - logiczne

- sterujące
- warunkowe
- specjalne
- typ danych
 - stałoprzecinkowe
 - zmiennoprzecinkowe
 - bitowe
 - blokowe
 - DSP - Digital Signal Processing
 - SIMD - Single Instruction Multiple Data
- poziom uprzywilejowania

6.3 Podział mikroprocesorów ze względu na listę instrukcji

- RISC (ang. Reduced Instruction Set Computer)
 - zbiór instrukcji jest ortogonalny
 - mała liczba instrukcji
- CISC (ang. Complex Instruction Set Computer)
 - rozbudowana liczba instrukcji
 - wysoka specjalizacja instrukcji

6.4 Instrukcje przesłań

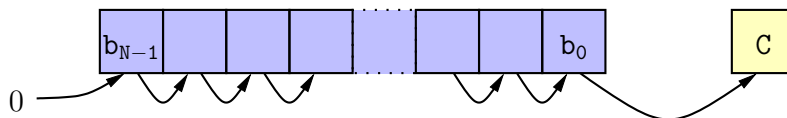


$$[AD] = RS$$

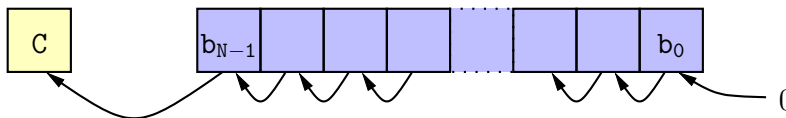
$$RD_H \circ RD_L = [AS] \circ [AS + 1]$$

6.5 Instrukcje przesunięcia

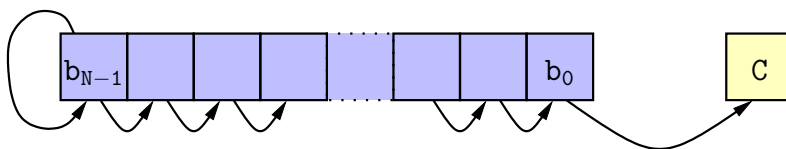
- logiczne w prawo



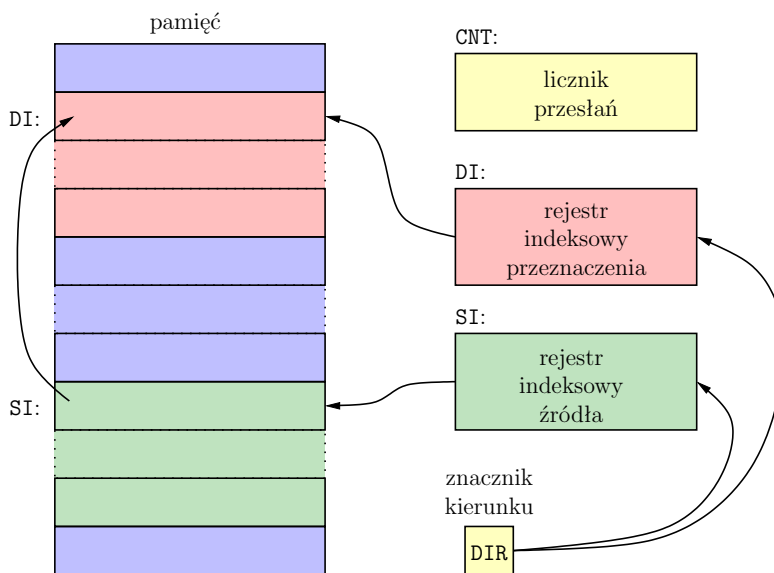
- w lewo



- arytmetyczne w prawo



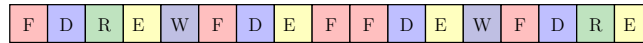
6.6 Instrukcje blokowe



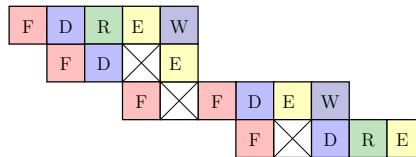
7 Układ wykonawczy

7.1 Cykl wykonywania instrukcji

- przetwarzanie sekwencyjne



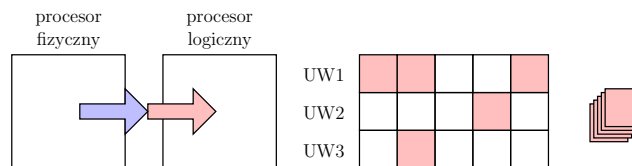
- przetwarzanie potokowe



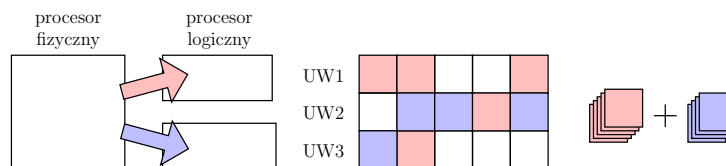
F - pobranie kodu instrukcji
 D - dekodowanie instrukcji
 R - odczyt argumentów
 E - wykonanie instrukcji
 W - zapisanie wyniku
 X - oczekiwanie

7.2 Wielowątkowość

- Standardowy procesor

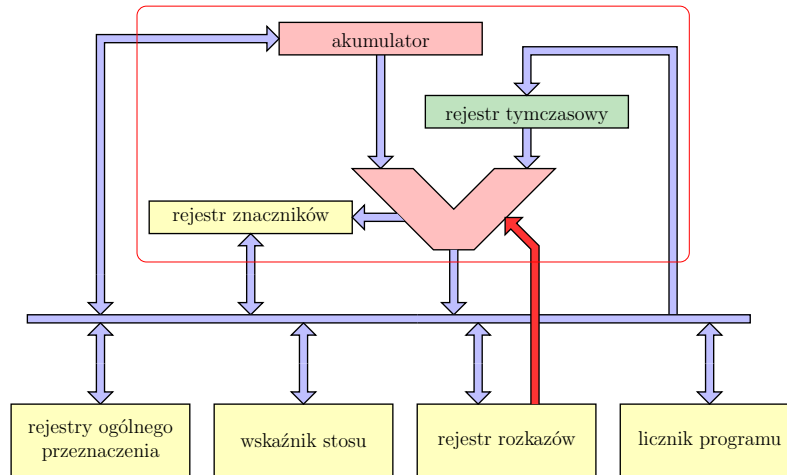


- Procesor z technologią HT

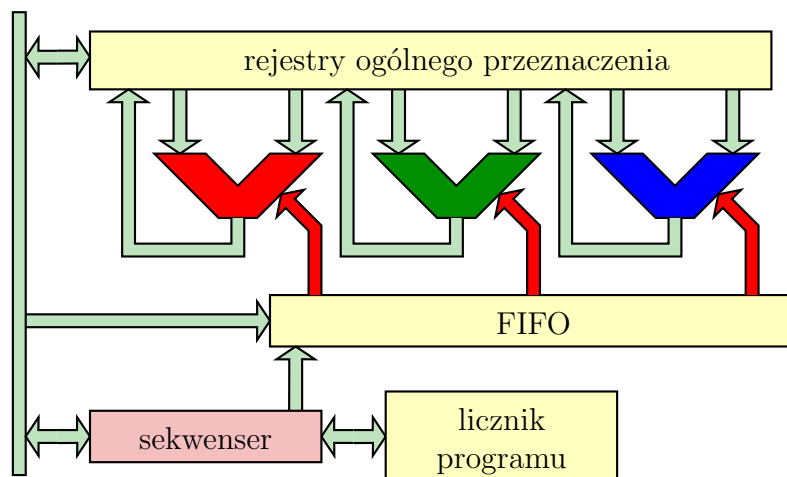


7.3 Hiperskalarność

- Standardowy układ wykonawczy

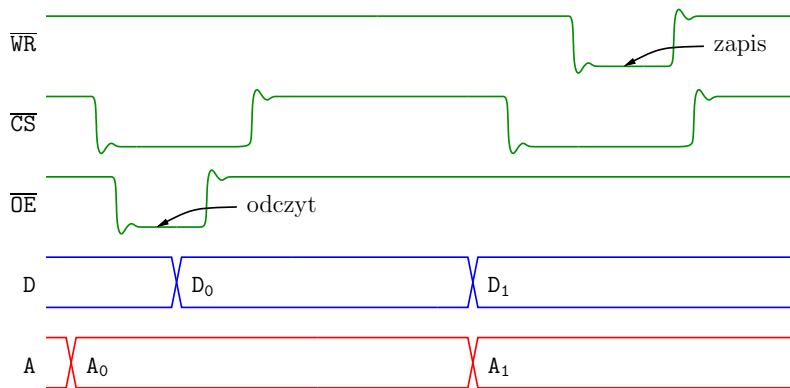


- Hiperskalarny układ wykonawczy

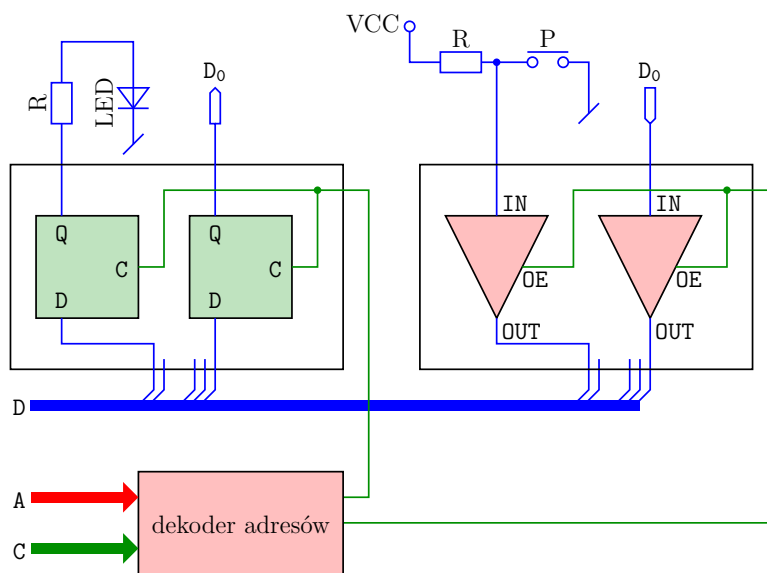


8 Urządzenia peryferyjne

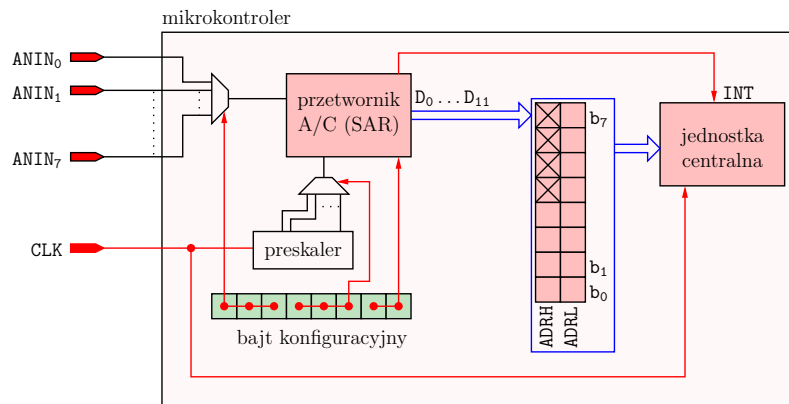
8.1 Magistrale



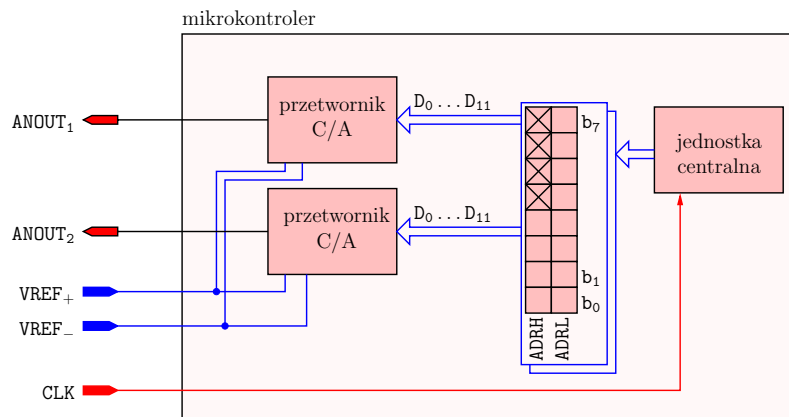
8.2 Porty wejścia wyjścia



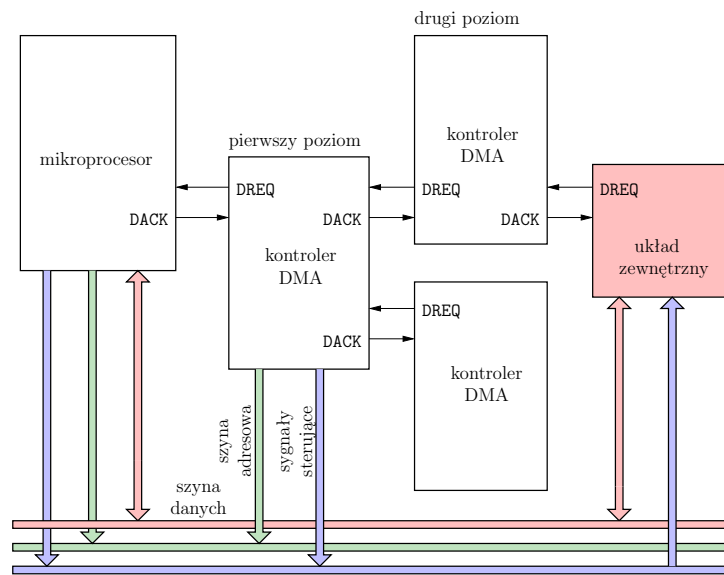
8.3 Przetwornik analogowo cyfrowy



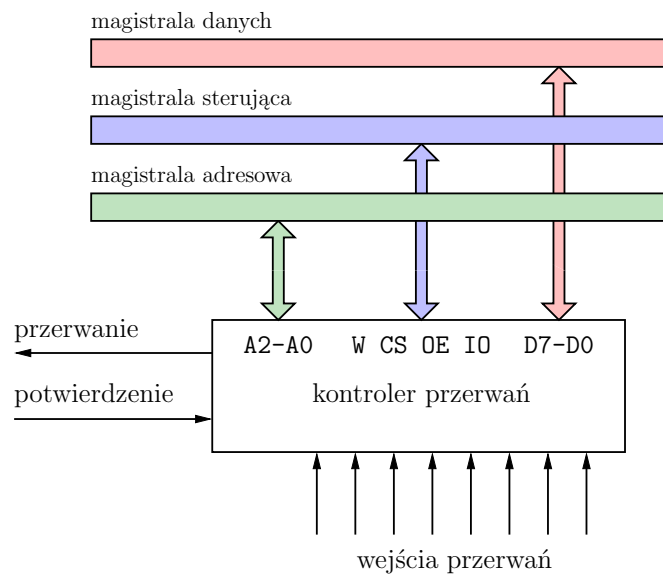
8.4 Przetwornik cyfrowo analogowy

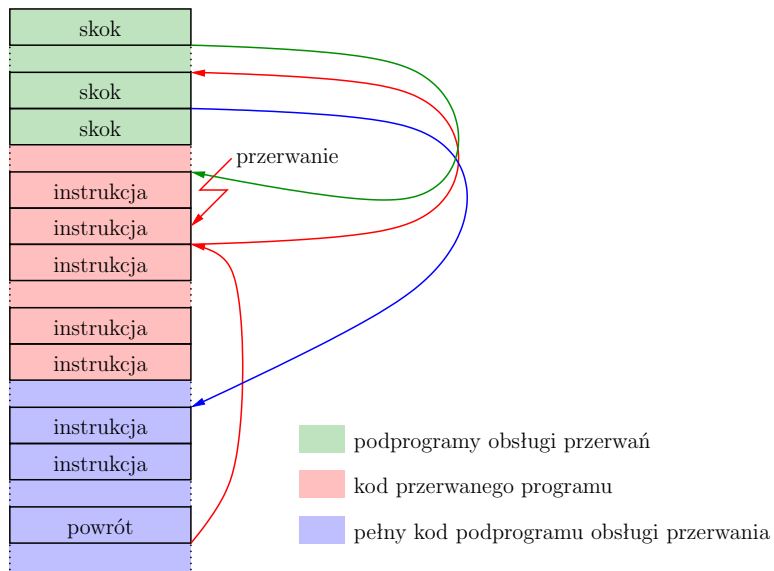


8.5 Bezpośredni dostęp do pamięci



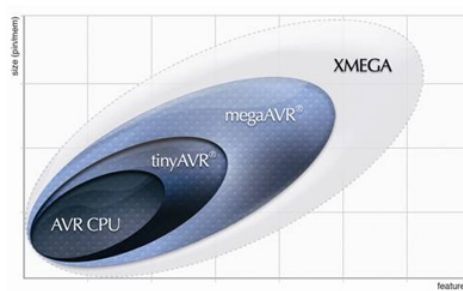
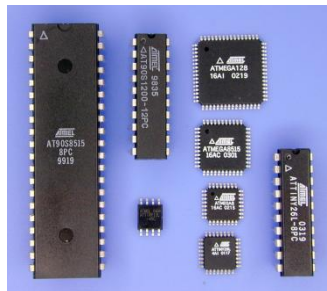
8.6 Kontroler przerwań





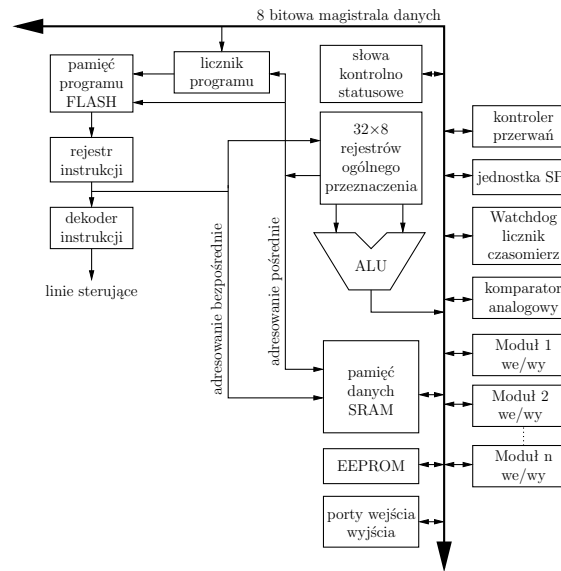
9 8 bitowe mikrokontrolery rodziny AVR

9.1 Podział



- tinyAVR
- megaAVR
- XMEGA

9.2 Rdzeń AVR



9.3 Rejestr statusowy

numer bitu	7							0
nazwa	I	T	H	S	V	N	Z	C
dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
wartość początkowa	0	0	0	0	0	0	0	0

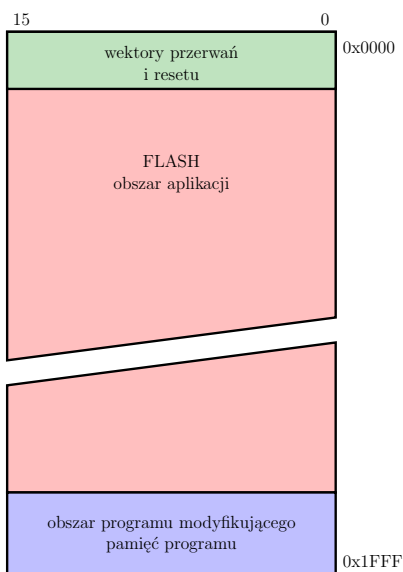
- I - globalne zezwolenie na przerwania
- T - znacznik kopii
- H - znacznik przeniesienia połówkowego
- S - bit znaku $S = V \oplus N$
- V - znacznik przepełnienia
- N - znacznik wartości ujemnej
- Z - znacznik wartości zerowej
- C - znacznik przeniesienia lub pożyczki

9.4 Rejestry ogólnego przeznaczenia

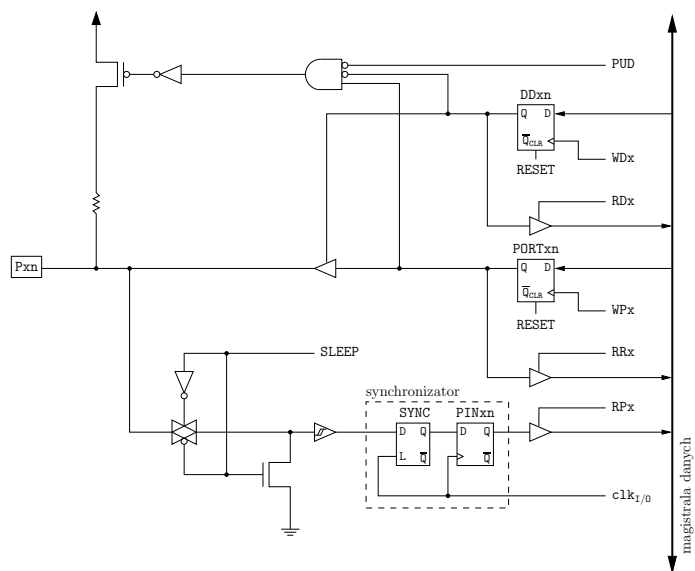
7	0	adres	
R0		0x00	
R1		0x01	
R2		0x02	
...			
R13		0x0D	
R14		0x0E	
R15		0x0F	
R16		0x10	
R17		0x11	
...			
R26		0x1A	młodszy bajt rejestru X
R27		0x1B	starszy bajt rejestru X
R28		0x1C	młodszy bajt rejestru Y
R29		0x1D	starszy bajt rejestru Y
R30		0x1E	młodszy bajt rejestru Z
R31		0x1F	starszy bajt rejestru Z

X, Y, Z - rejestry indeksowe/wskaźnikowe

9.5 Mapa pamięci programu



9.6 Budowa cyfrowego portu wejścia wyjścia

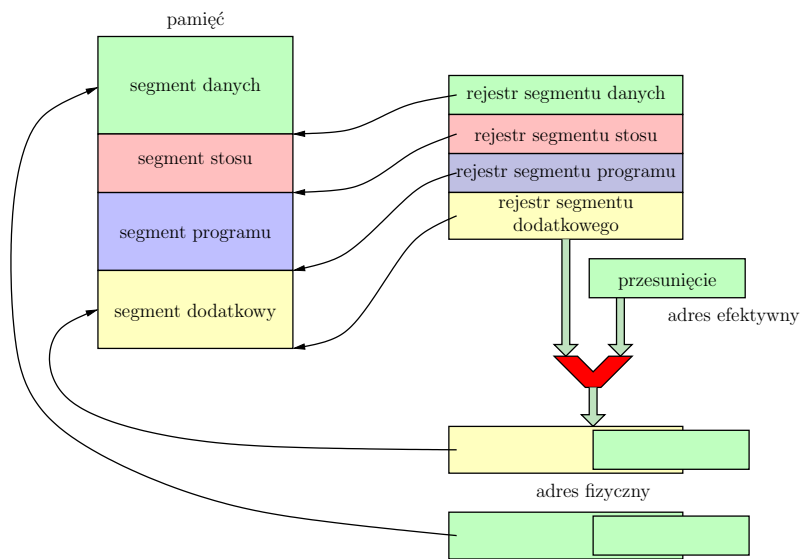


9.7 Przykładowe instrukcje

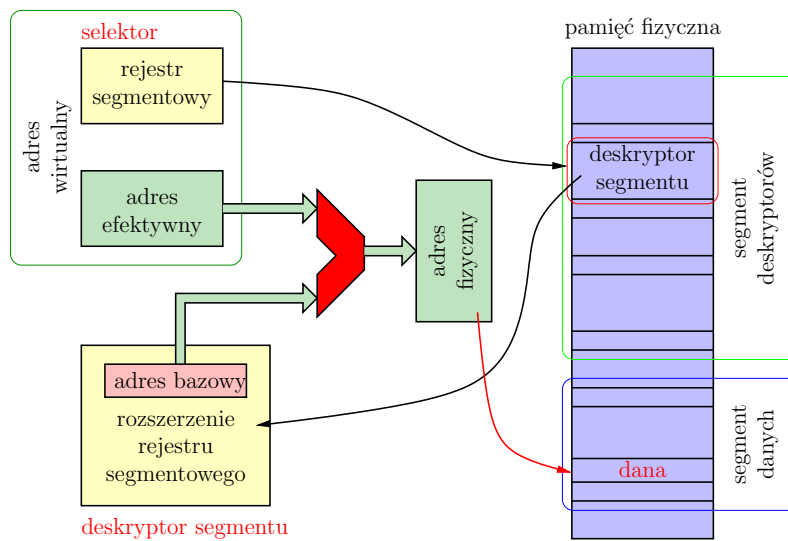
mnemonik	operandy	opis	znaczniki	#	uwagi
ADD	Rd, Rr	$Rd = Rd + Rr$	ZCNVSH	1	$d, r = [0, 31]$
ADC	Rd, Rr	$Rd = Rd + Rr + C$	ZCNVSH	1	$d, r = [0, 31]$
ADIW	Rd, K	$Rd + 1: Rd = Rd + 1: Rd + K$	ZCNVSH	2	$K = [0, 63], d = 24, 26, 28, 30$
COM	Rd	$Rd = 0xFF - Rd$	ZCNVSH	1	$d = [0, 31]$
NEG	Rd	$0x00 - Rd$	ZCNVSH	1	$d = [0, 31]$
CALL	k	$[SP] = PC + 2, PC = k$	-	4/5	$k = [0, 4M - 1]$
RET	-	$PC = [SP]$	-	4/5	-
RETI	-	$I = 1, PC = [SP]$	I	4/5	-
BREQ	k	if(Z==1) $PC = PC + 1 + k$	-	1/2	$k = [-64, 63]$
MOV	Rd, Rr	$Rd = Rr$	-	1	$r, d = [0, 31]$
LDI	Rd, K	$Rd = K$	-	1	$d = [16, 31], K = [0, 255]$
LDS	Rd, k	$Rd = [k]$	-	2	$d = [0, 31], k = [0, 64k - 1]$
LD	Rd, X	$Rd = [X]$	-	2	$d = [0, 31]$
LPM	-	$R0 = [Z]$	-	3	-

10 Procesory rodziny x86

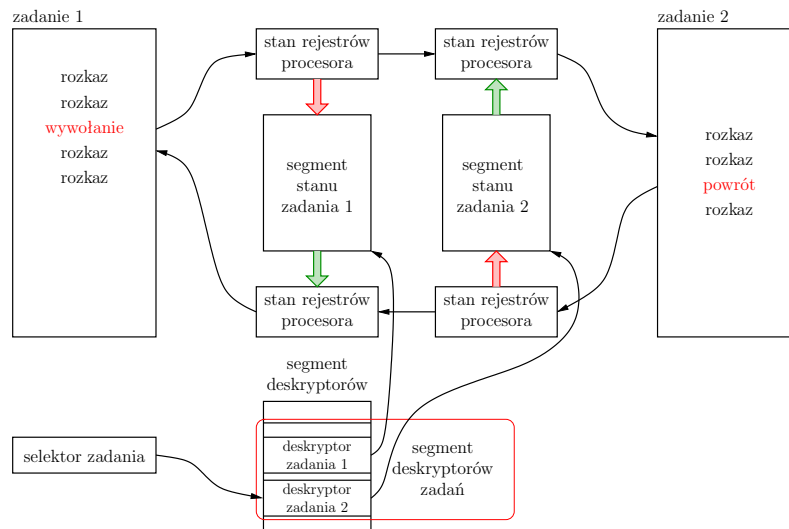
10.1 Adresowanie w trybie rzeczywistym



10.2 Adresowanie w trybie chronionym



10.3 Przełączanie zadań



11 Literatura

- [1] 8-bit Microcontroller with 16K Bytes In-System Programmable Flash - ATmega16 ATmega16L. Atmel 2006.
- [2] Mikrokontrolery AVR w praktyce. Jarosław Doliński. BTC. Warszawa 2003.
- [3] Intel386 TM DX MICROPROCESSOR 32-BIT CHMOS MICROPROCESSOR WITH INTEGRATED MEMORY MANAGEMENT. Intel. 1995.
- [4] IA-32 Intel Architecture Software Developer's Manual Volume 1-3. Intel. 2004.
- [5] Mikrokomputery klasy IBM PC. Henryk Małyśiak, Bolesław Pochopień, Eugeniusz Wróbel. WNT. Warszawa 1992.
- [6] Anatomia PC. Piotr Metzger, Adam Jelowicki. Helion. Wydanie czwarte lub nowsze. 1998.