

Systemy mikroprocesorowe

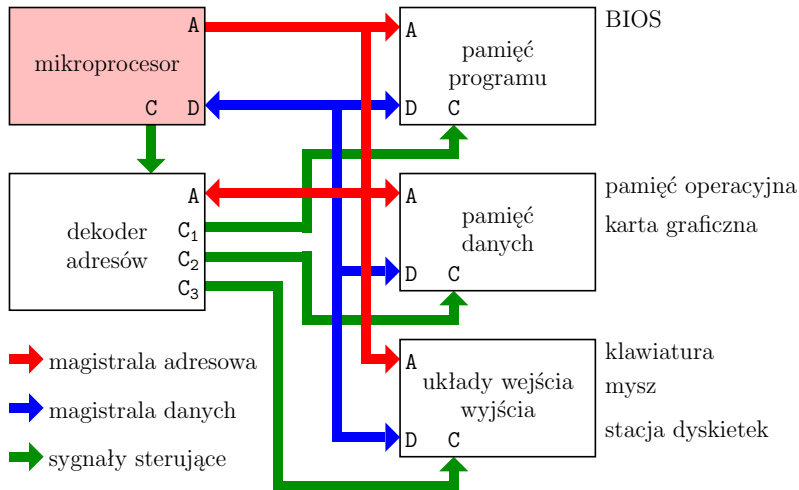
Dariusz Chaberski

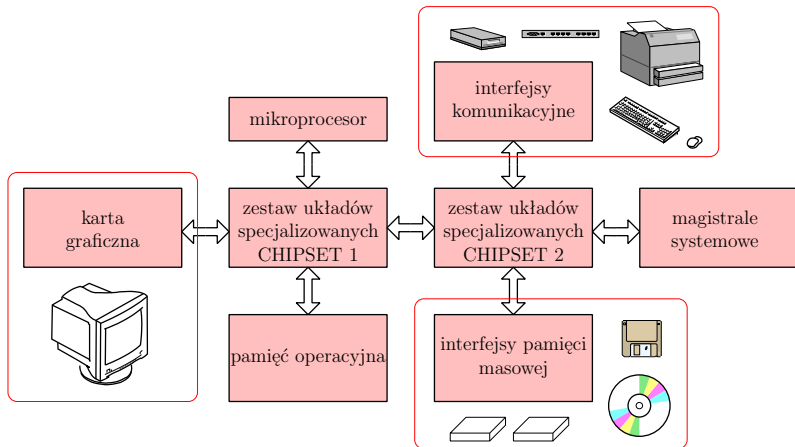
Uniwersytet Mikołaja Kopernika w Toruniu
Wydział Fizyki, Astronomii i Informatyki Stosowanej

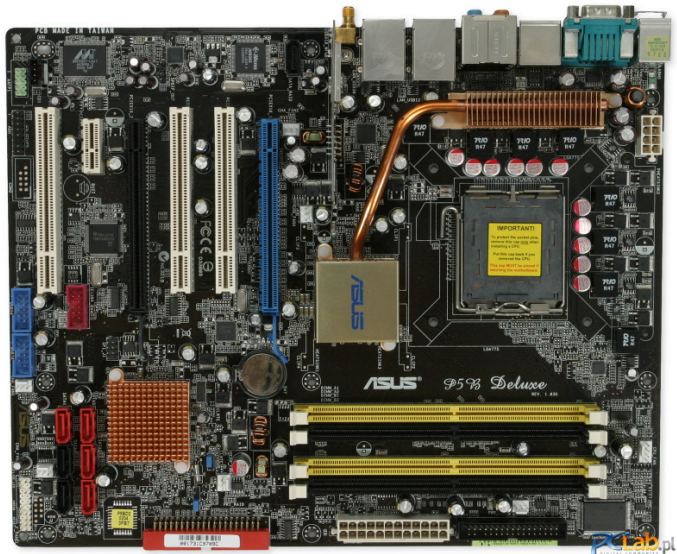
Regionalne Kółka Fizyczne
Urząd Marszałkowski w Toruniu
Program Operacyjny Kapitał Ludzki

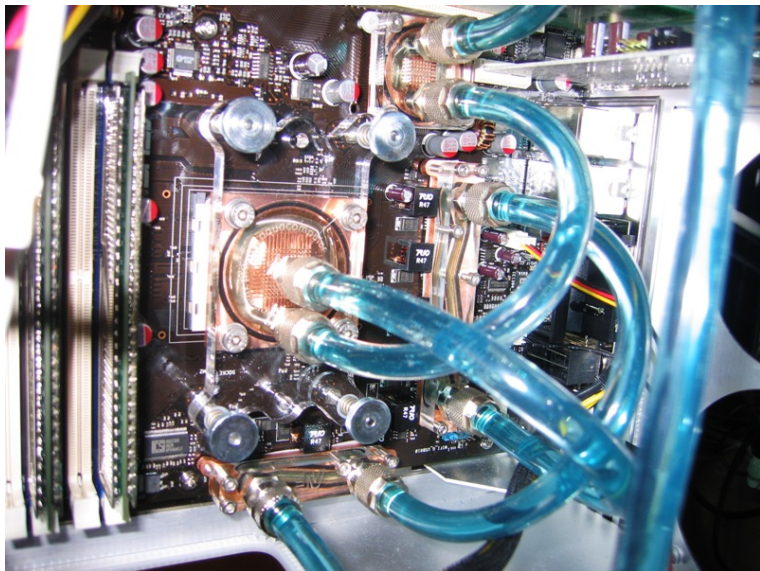
Grudziądz, 3 października 2009

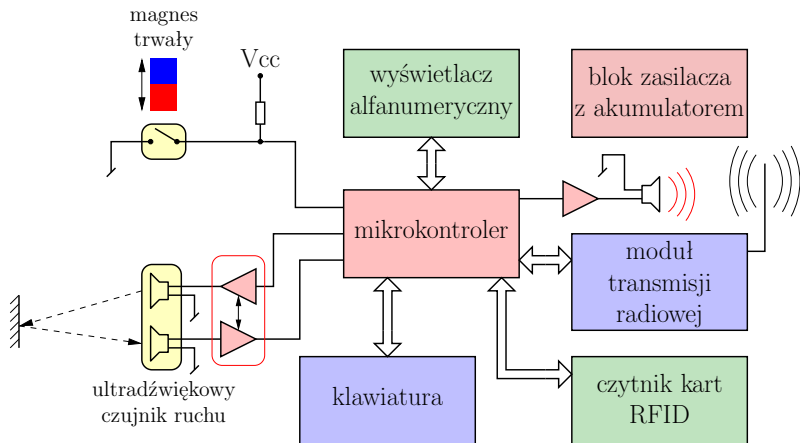
- email:
rkf.chaberski@fizyka.umk.pl
- www:
<http://www.fizyka.umk.pl/~rkf.chaberski>
<http://www.fizyka.umk.pl/rkf/RKFGrudziadz.html>
- 20 godzin = 2 godziny + 6 × 3 godziny
 - Październik 3, 10, 17 24, 31
 - Listopad 7, 14, 21, 28
 - Grudzień 5, 12, 19, 26
 - Styczeń 2, 9, 16

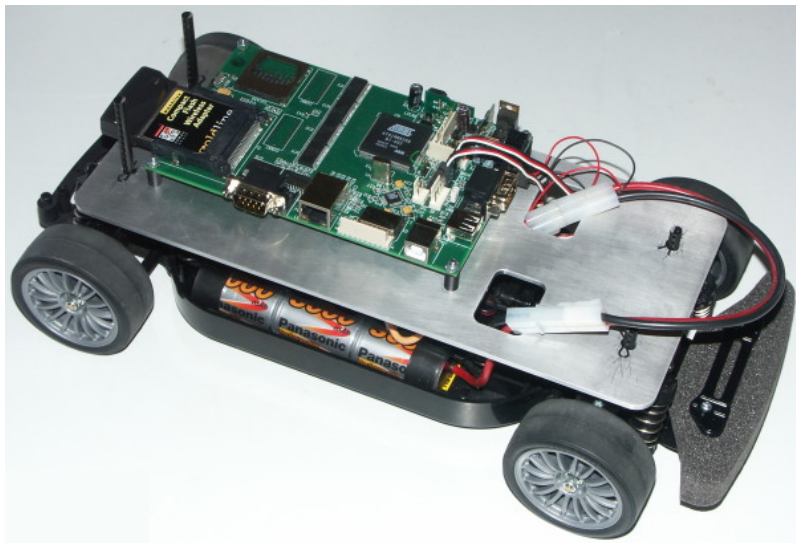


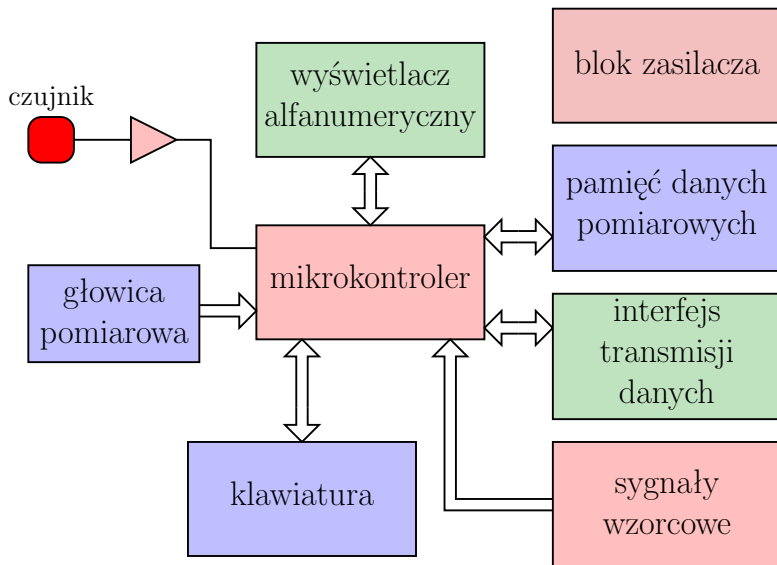


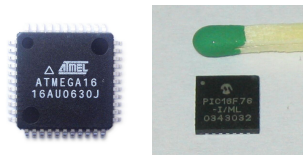












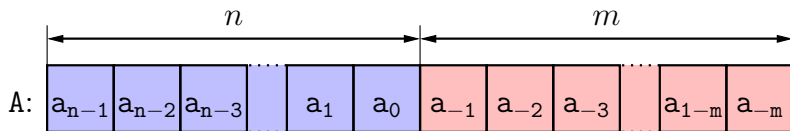
- mikrokontroler

- 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

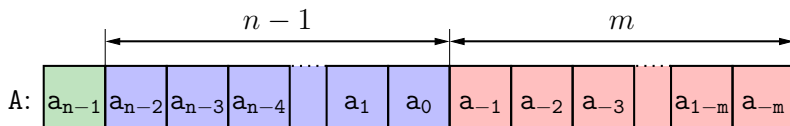


- mikroprocesor

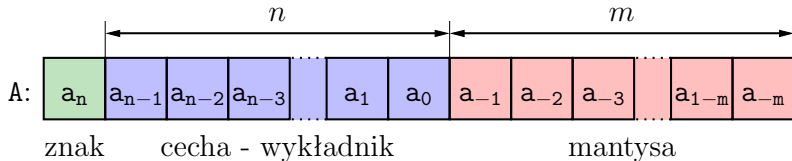
- duża moc obliczeniowa (10000 MIPSów obecnie)
- wielordzeniowość oraz technologia HT
- przetwarzanie wielopotokowe
- dostępne instrukcje zmiennoprzecinkowe
- do działania wymaga dodatkowych peryferii (kontroler DMA, pamięć (programu i danych), kontroler przerwań)
- dostępny układ zarządzania pamięcią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



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



$$-L(A) = L(\bar{A})$$



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

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

$$S = A[n]$$

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

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

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

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

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

A:

1	1	0	1	0	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---

+

B:

0	0	1	1	0	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---

=

A+B:

0	0	0	0	1	0	0	1	1	0
---	---	---	---	---	---	---	---	---	---

C:

1

 Z:

0

 N:

0

 V:

0

 S:

0

 H:

0

A:

1	1	0	1	0	1	1	1
---	---	---	---	---	---	---	---

— B:

1	1	1	0	1	0	0	0
---	---	---	---	---	---	---	---

— A-B:

1	1	1	0	1	1	1	1
---	---	---	---	---	---	---	---

C:

1

 Z:

0

 N:

1

 V:

0

 S:

1

 H:

1

A:

1	1	0	1	0	1	1	1
---	---	---	---	---	---	---	---

\bar{B} :

0	0	0	1	0	1	1	1
---	---	---	---	---	---	---	---

+

1

=

A-B:

1	1	1	0	1	1	1	1
---	---	---	---	---	---	---	---

\bar{C} :

0

 Z:

0

 N:

1

 V:

0

 S:

1

 \bar{H} :

0

A:

1	1	0	1	0	1	1	1
---	---	---	---	---	---	---	---

×

B:

1	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---

=

1	1	0	1	0	1	1	1
---	---	---	---	---	---	---	---

1	1	0	1	0	1	1	1
---	---	---	---	---	---	---	---

+

1	1	0	1	0	1	1	1
---	---	---	---	---	---	---	---

=

A×B:

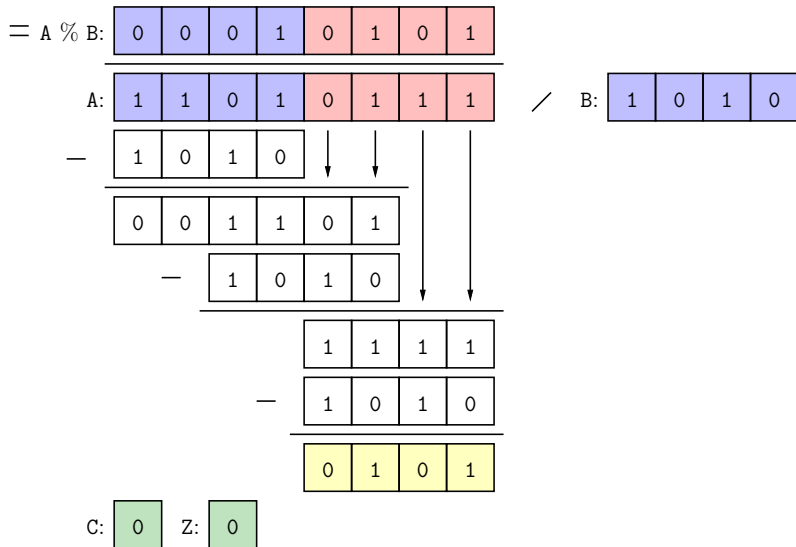
1	1	0	1	1	1	1	1	1	0	1	1	0	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

C:

0

Z:

0



A:

0	0	0	1	0	1	0	1
---	---	---	---	---	---	---	---

B:

1	1	0	1	0	1	1	1
---	---	---	---	---	---	---	---

$A \wedge B$:

0	0	0	1	0	1	0	1
---	---	---	---	---	---	---	---

$A \vee B$:

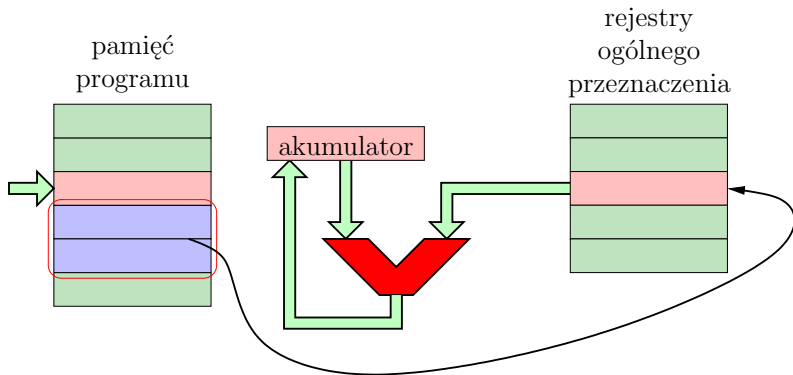
1	1	0	1	0	1	1	1
---	---	---	---	---	---	---	---

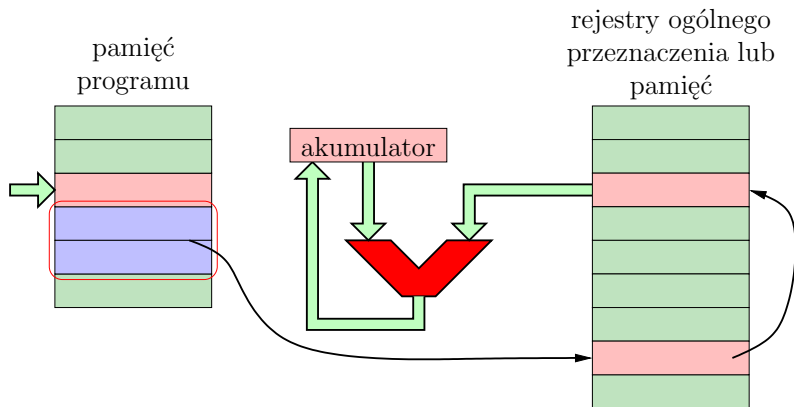
$A \oplus B$:

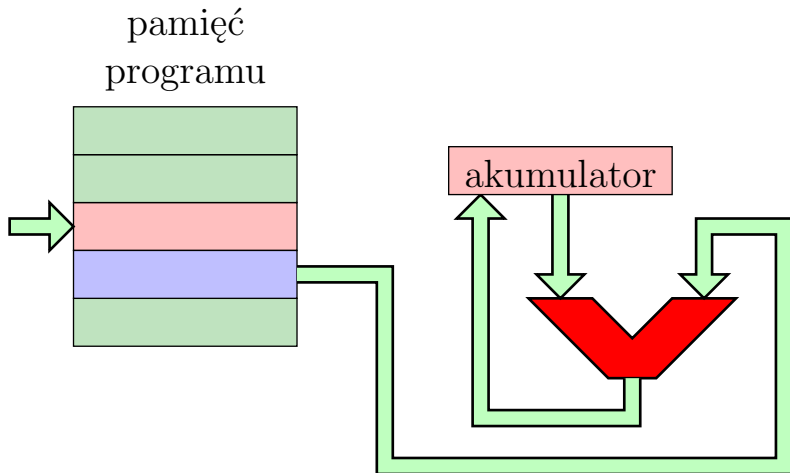
1	1	0	0	0	0	1	0
---	---	---	---	---	---	---	---

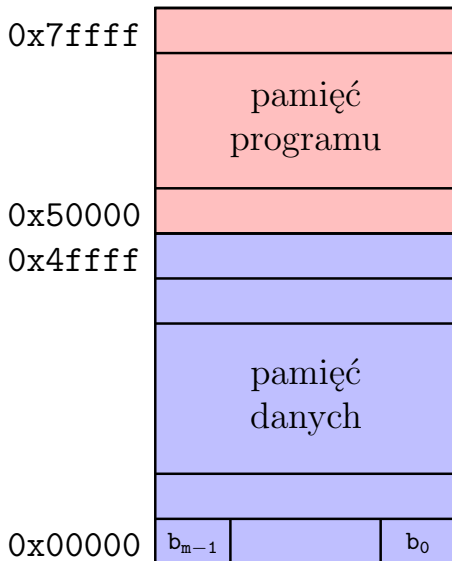
\bar{A} :

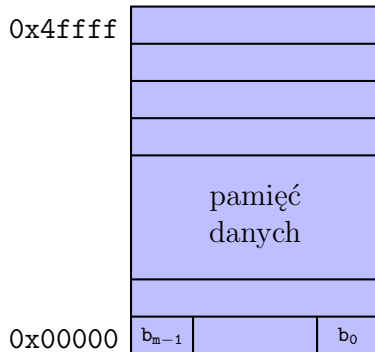
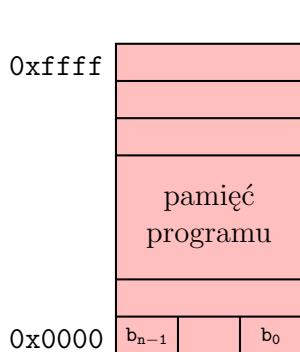
1	1	1	0	1	0	1	0
---	---	---	---	---	---	---	---

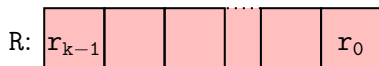




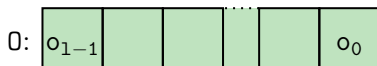




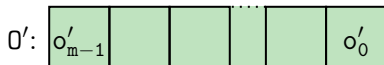




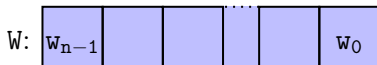
rodzaj operacji



pierwszy operand



drugi operand



wynik

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

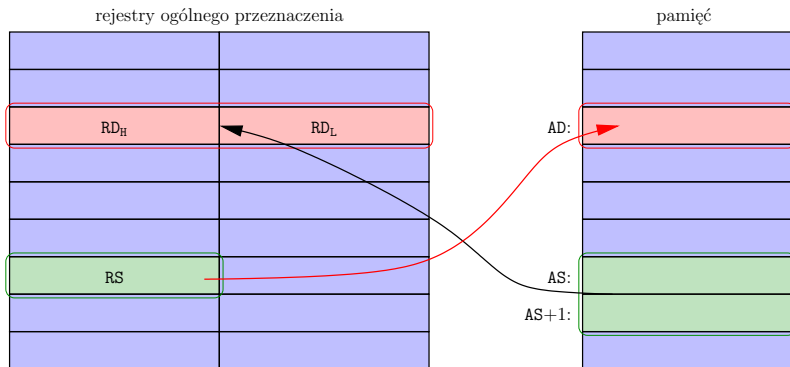
- przykłady

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

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

- 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

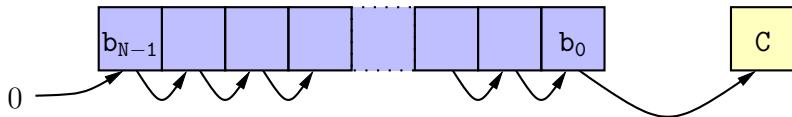
- RISC (ang. Reduced Instruction Set Computer)
 - zbiór instrukcji jest ortogonalny
 - mała liczba instrukcji
- CISC (ang. Complex Instruction Set Computer)
 - rozbudowana liczba instrukcji
 - wysoka specjalizacja instrukcji



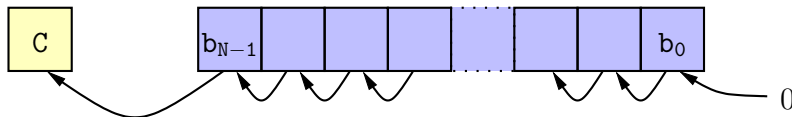
$$[AD] = RS$$

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

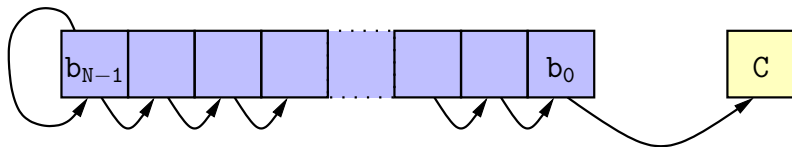
- logiczne w prawo

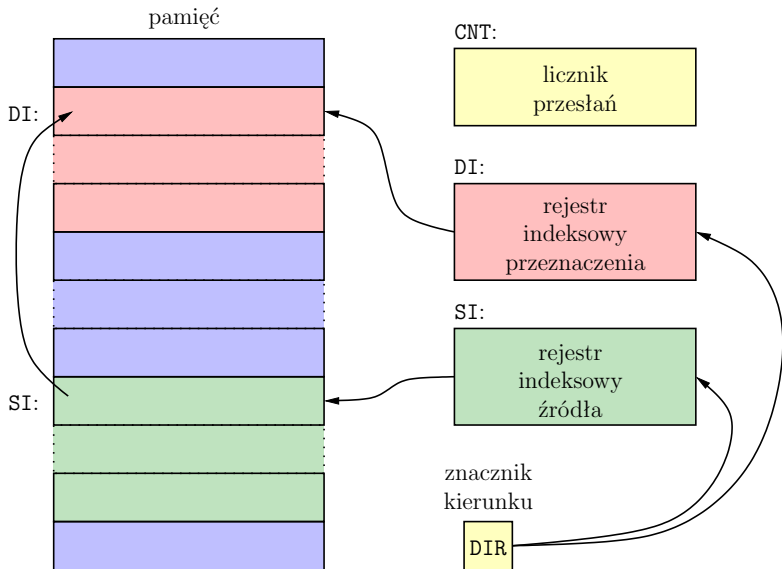


- w lewo

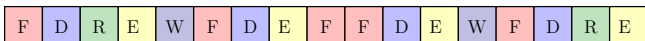


- arytmetyczne w prawo

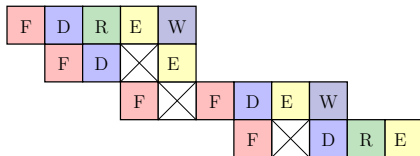




- przetwarzanie sekwencyjne



- przetwarzanie potokowe



F - pobranie kodu instrukcji

D - dekodowanie instrukcji

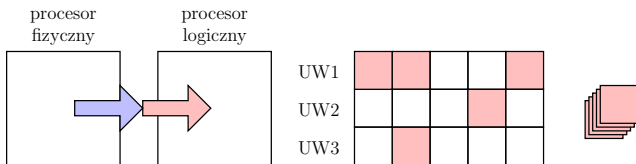
R - odczyt argumentów

E - wykonanie instrukcji

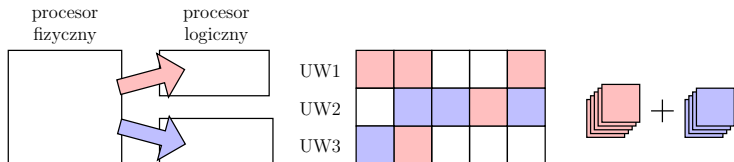
W - zapisanie wyniku

× - oczekiwanie

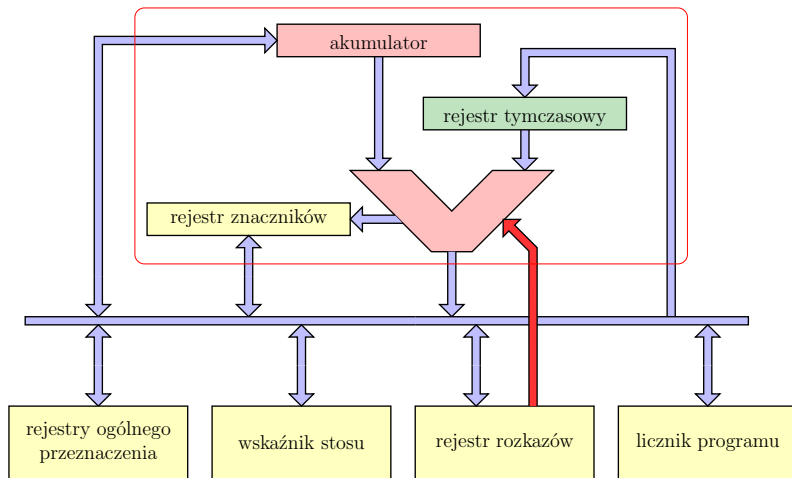
- Standardowy procesor



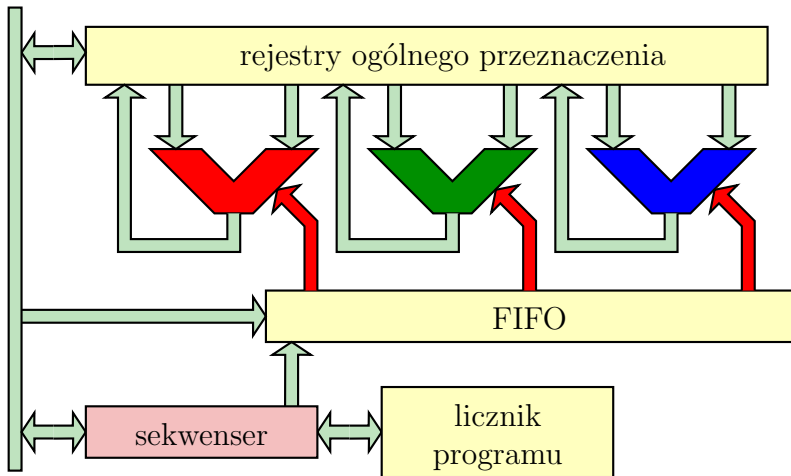
- Procesor z technologią HT

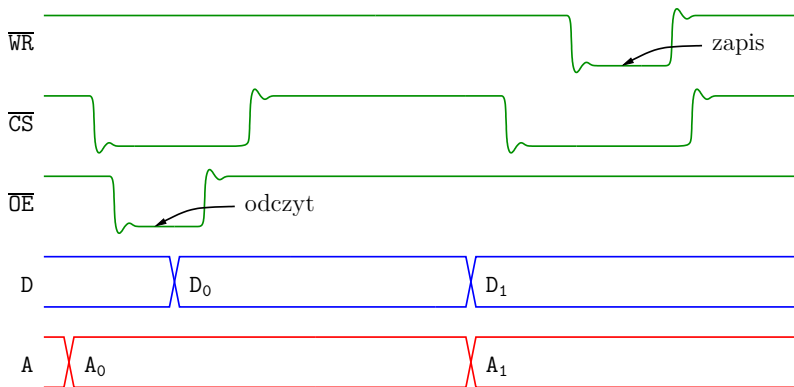


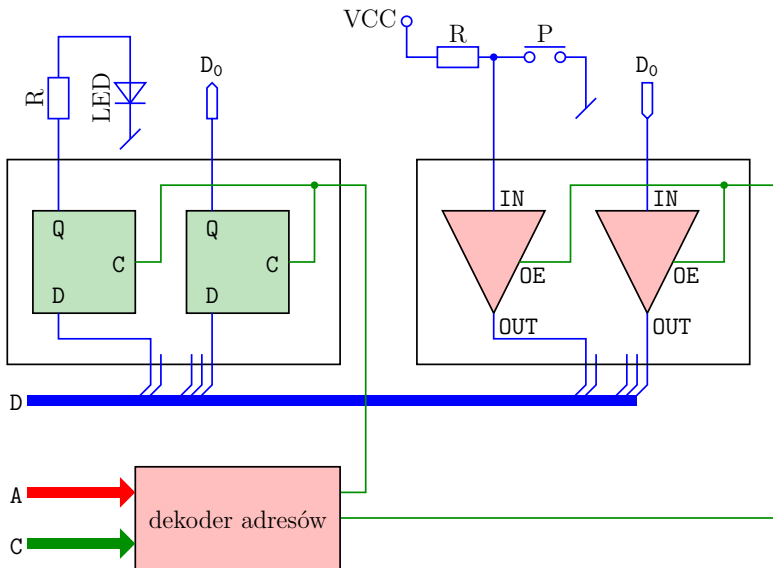
- Standardowy układ wykonawczy

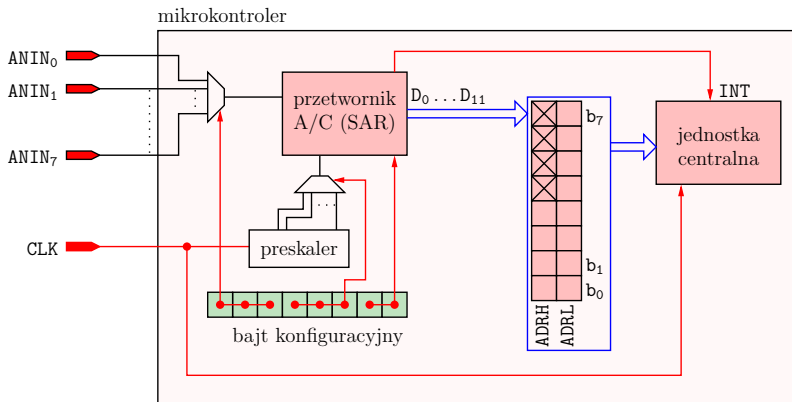


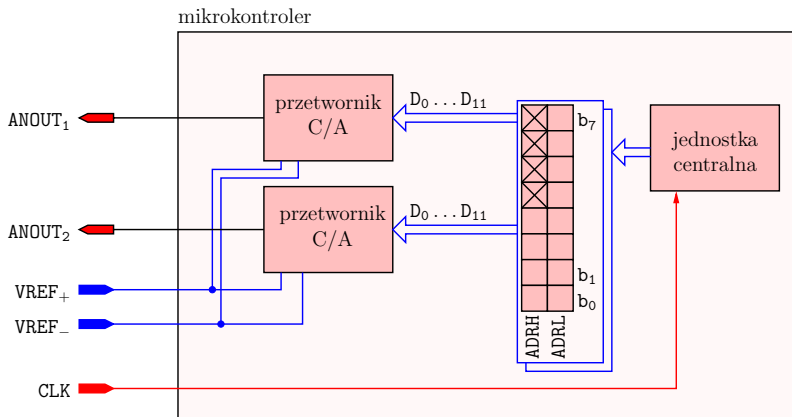
- Hiperskalarny układ wykonawczy

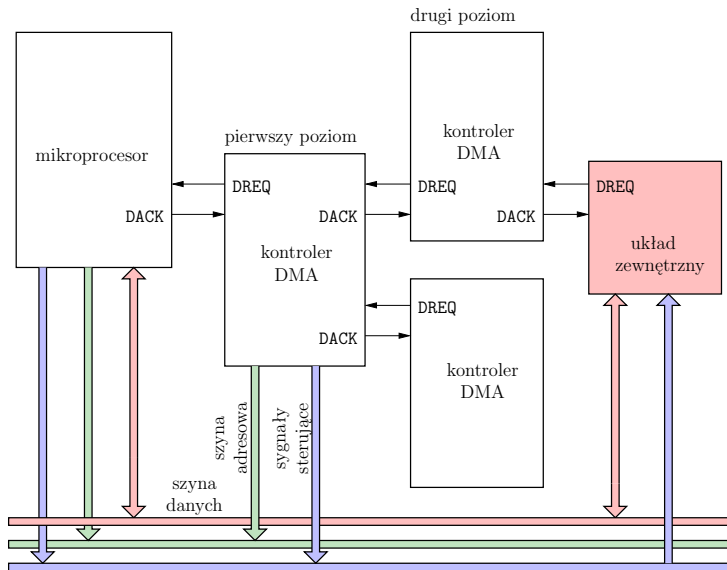


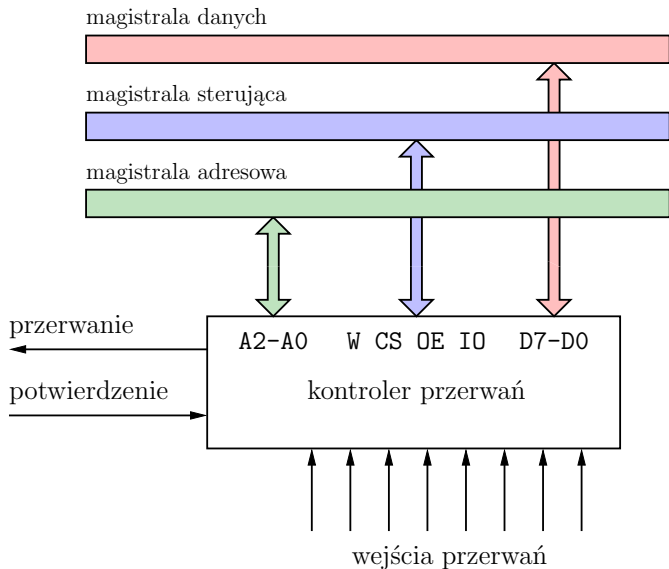


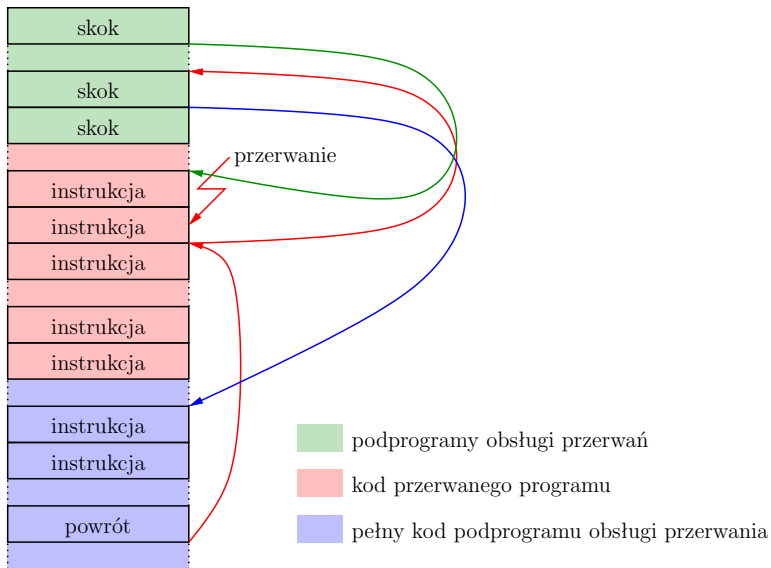


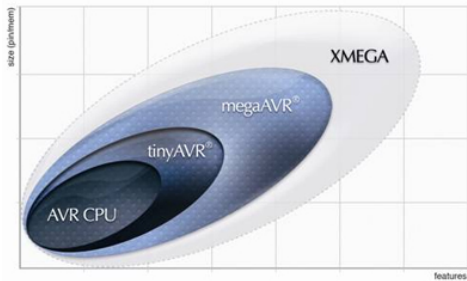




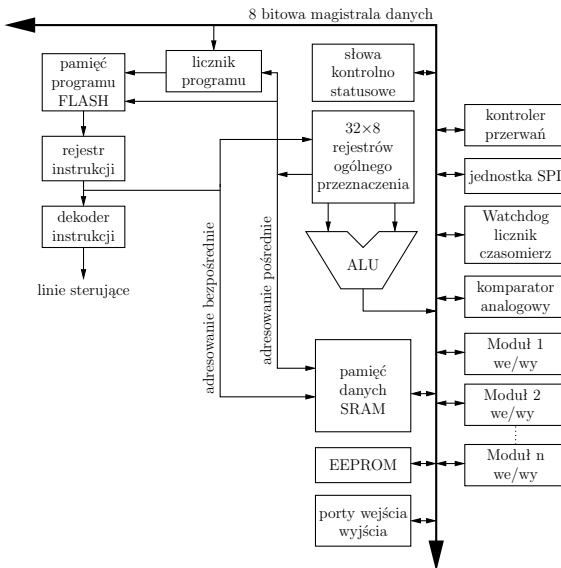








- tinyAVR
- megaAVR
- XMEGA



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

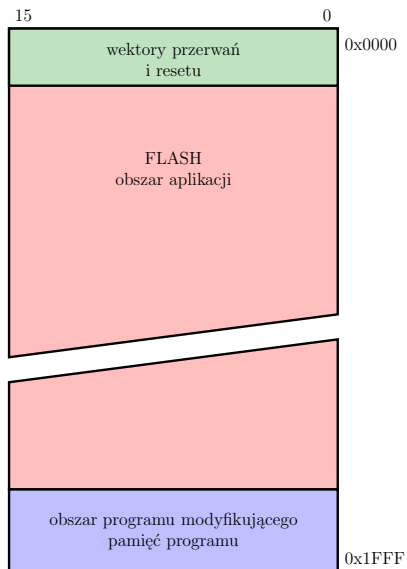
N - znacznik wartości ujemnej

Z - znacznik wartości zerowej

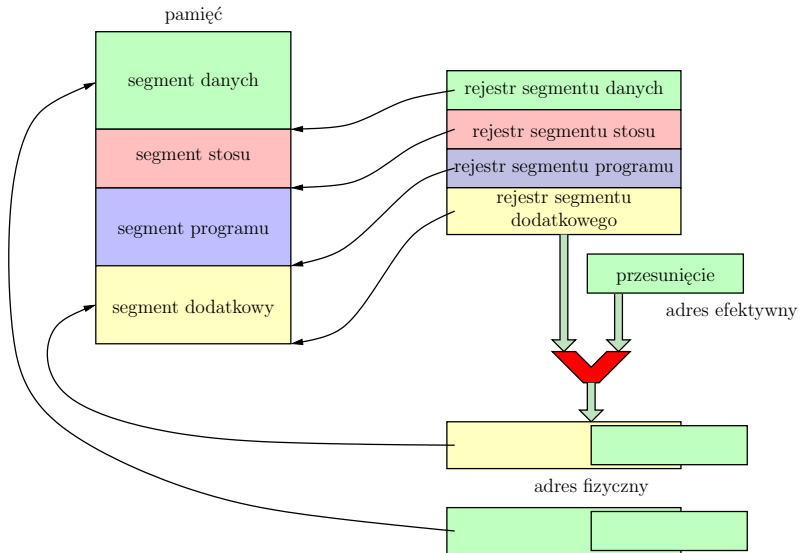
C - znacznik przeniesienia lub pożyczki

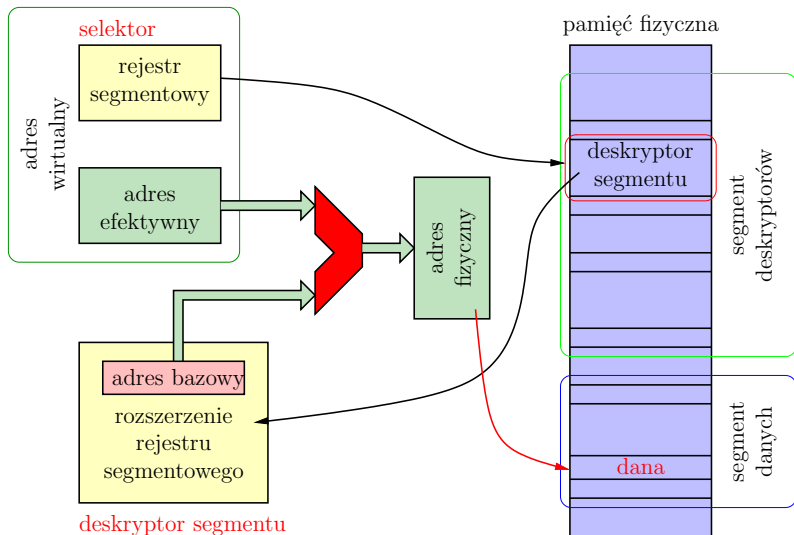
7	0	adres	
R0		0x00	
R1		0x01	
R2		0x02	
...			
R13		0x0D	
R14		0x0E	
R15		0x0F	
R16		0x10	
R17		0x11	
...			
R26		0x1A	młodszy bajt rejestru X
R27		0x1B	starszy bajt rejestru X
R28		0x1C	młodszy bajt rejestru Y
R29		0x1D	starszy bajt rejestru Y
R30		0x1E	młodszy bajt rejestru Z
R31		0x1F	starszy bajt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

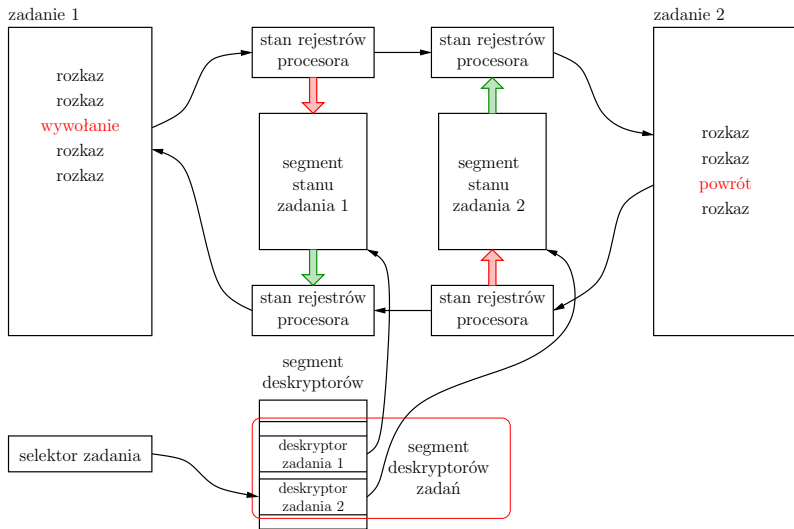
X, Y, Z - rejestry indeksowe/wskaźnikowe



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	$Rd + 1: Rd = Rd + 1: Rd + K$	ZCNVS	2	$K = [0, 63]$, $d = 24, 26, 28, 30$
COM	Rd	$Rd = 0xFF - Rd$	ZCNVS	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	-







- [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 Małysiak, Bolesław Pochopień, Eugeniusz Wróbel. WNT. Warszawa 1992.
- [6] Anatomia PC. Piotr Metzger, Adam Jełowicki. Helion. Wydanie czwarte lub nowsze. 1998.