



# Politechnika Wrocławska

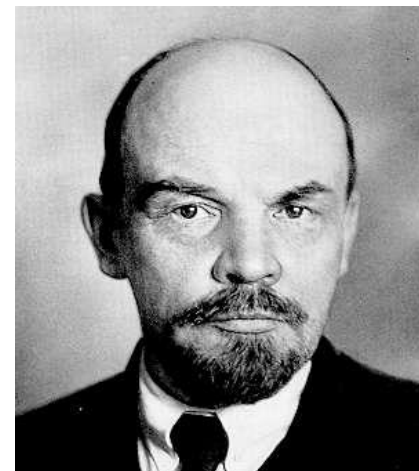
## **Podstawy Techniki Mikroprocesorowej**

### **wykład 2: procesor z siwą bródką i może coś jeszcze**

Dr inż. Jacek Mazurkiewicz  
Katedra Informatyki Technicznej  
e-mail: [Jacek.Mazurkiewicz@pwr.edu.pl](mailto:Jacek.Mazurkiewicz@pwr.edu.pl)

# „Kompjuter - eta jest’” i klasyfikacja

- jednostka centralna - procesor
- pamięć operacyjna
- urządzenia wejścia-wyjścia
- magistrale
  - von Neumann (!)
- SISD - Single Instruction Single Data
- SIMD - Single Instruction Multiple Data
- MISD - Multiple Instruction Single Data
- MIMD - Multiple Instruction Multiple Data





# Komputer - jak drużynowy



- są wśród nas wszędzie!
- ogólnego przeznaczenia
- specjalizowane, dedykowane

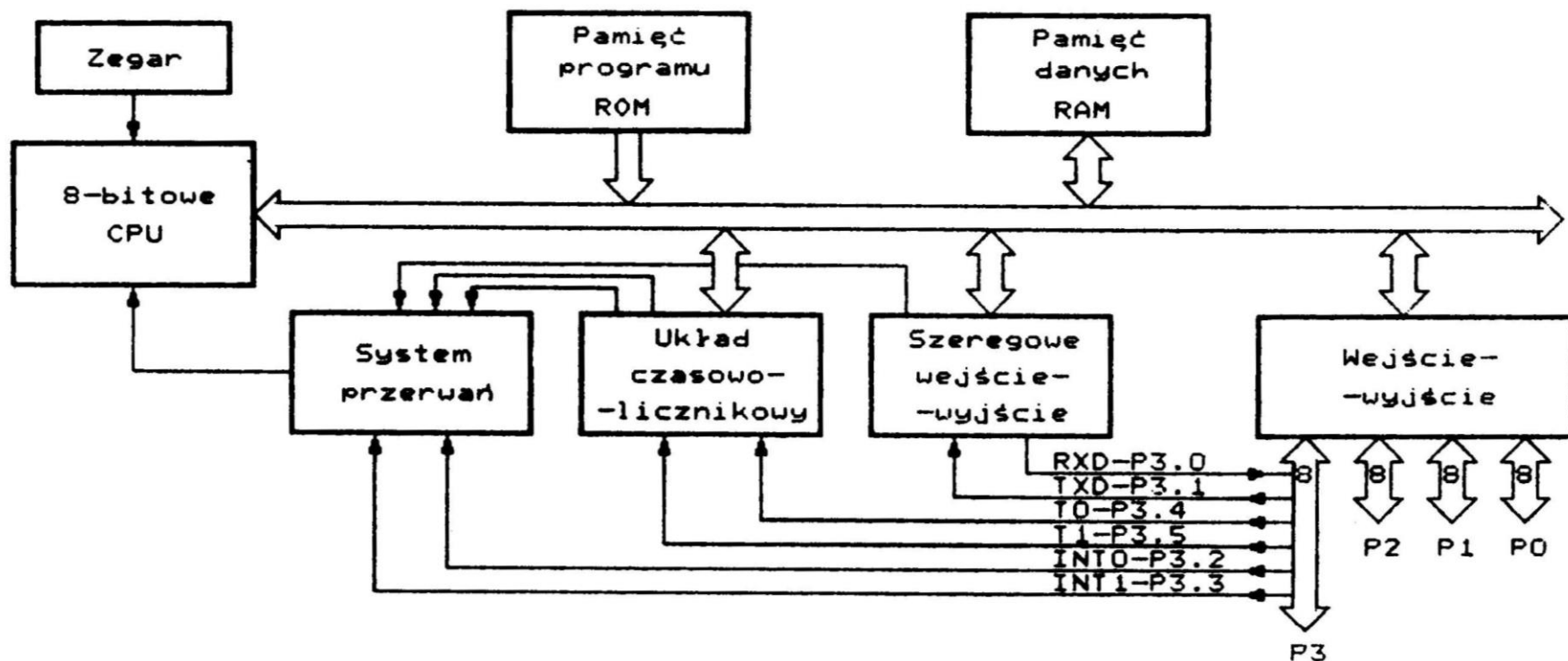




# To ja - INTEL się nazywam (8051)

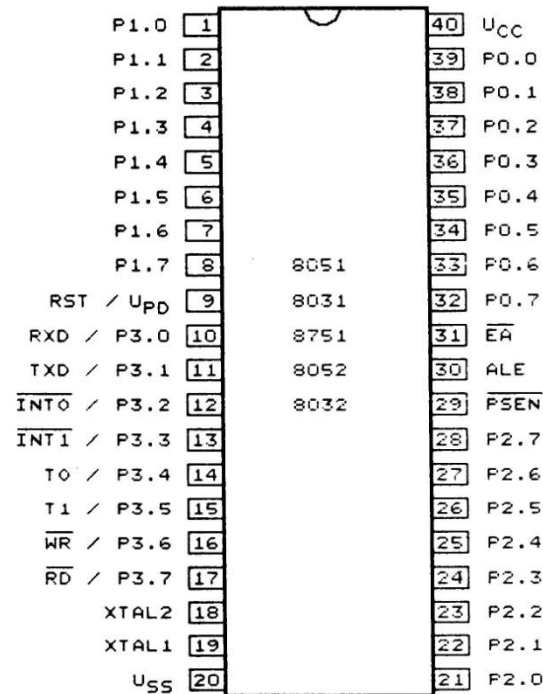
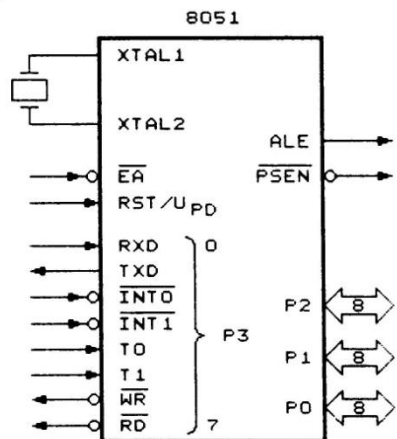
- Jednostka centralna:
  - 8 bitów, 111 rozkazów (CISC): 49 - 1 bajt, 45 - 2 bajty, 17 - 3 bajty)
  - 1 lub 2 cykle maszynowe, 4 cykle - mnożenie i dzielenie
- Zegar: wewnętrzny rezonator kwarcowy max 12 MHz - cykl maszynowy 1  $\mu$ s
- Wewnętrzna pamięć programu - ROM - 4KW (4096) 8-bit
  - może być i 8KW
- Wewnętrzna pamięć danych - RAM - 128B, zewnętrzna do 64KW
  - może być i 256B
- CTC: 2 16-bit liczniki, sterowane wewnętrznie lub zewnętrznie, 4 tryby pracy
  - może być licznik bonusowy!
- PS: nadawanie i odbiór niezależne, 4 tryby
- PR: 4 sztuki, po 8 linii, na nich też funkcje specjalne spoczywają
- Przerwania - 2-poziomowe, obsługują integralne elementy dodatkowe
- Żadne tam DMA!

# A tak ładnie wyglądam bez pokrywy



# Chcę oglądać twoje nogi - nogi - nogi

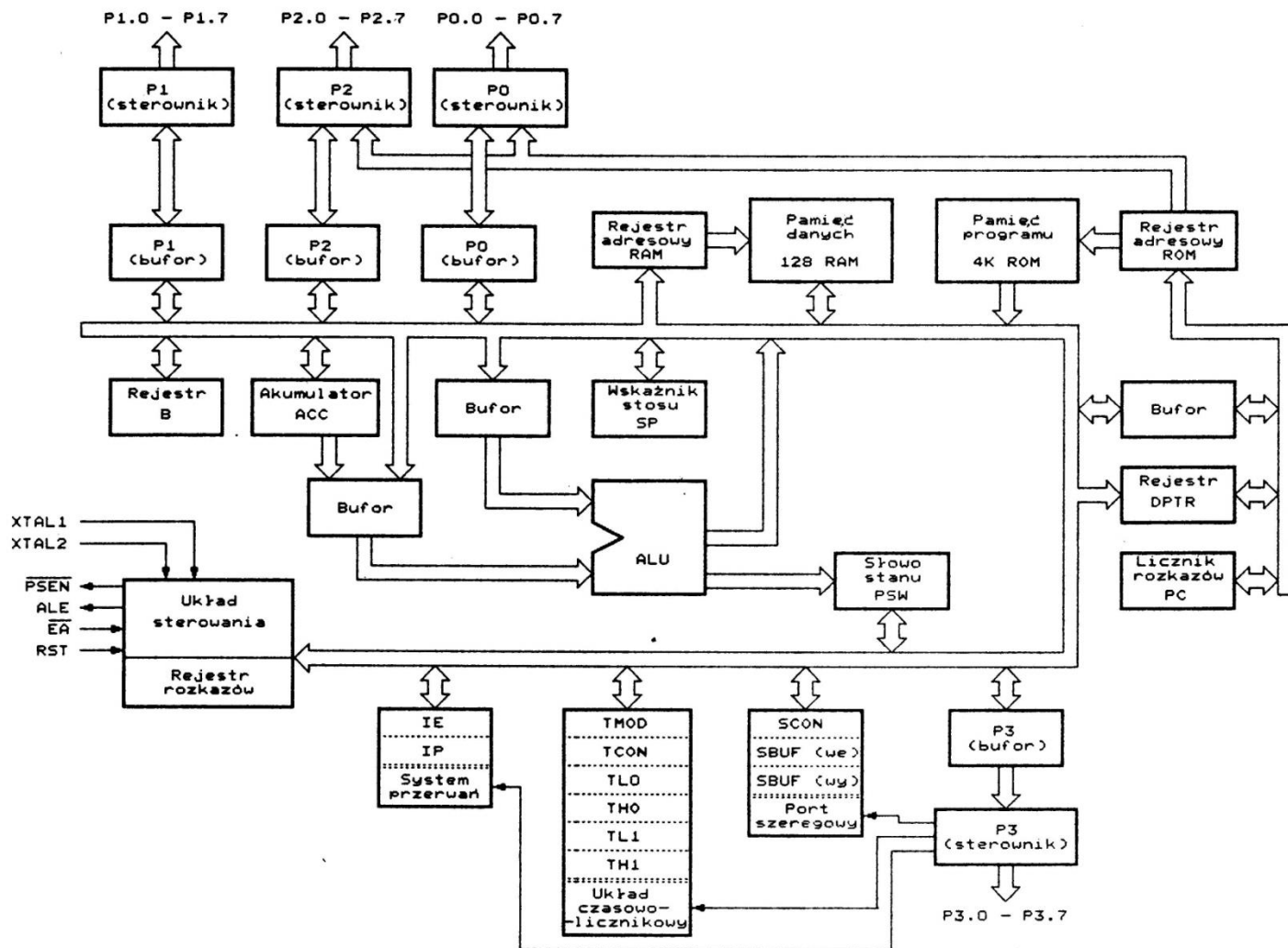
- P0 - i port i magistrala danych oraz adresowa młodsza
- P1 - i port i sterowanie T2
- P2 - i port i magistrala adresowa starsza
- P3 - i port i obsługa PS, T0, T1, INT0, INT1, WR, RD
- PSEN - Program Store Enable  
- odczyt z zewnętrznej pamięci programu
- ALE - Address Latch Enable - P0 - adres czy też dane
- RST/VPD - reset zerem, zasilanie w low-power
- EA - External Access - pobranie wszystkich rozkazów z zewnętrznej pamięci programu
- XTAL - oscylator kwarcowy
- Ucc - zasilanie
- Uss - masa







# A teraz gala rejestrowa!



# A ALU mam bardzo sprytnie!



- Dodawanie z przeniesieniem, odejmowanie z pożyczką:
  - akumulator jeden składnik obsługuje
  - drugi składnik - wewnętrzna pamięć danych, argument bezpośredni
  - wynik w akumulatorze - uwaga na PSW
- Mnożenie i dzielenie 8-bit bez znaku:
  - akumulator A i B: mnożna/mnożnik, dzielna/dzielnik
  - wynik - mnożenie - B i A, dzielenie - A - wynik, B - reszta
- Inkrementacja/dekrementacja akumulatora i komórki pamięci wewnętrznej
- Korekcja BCD wyniku w akumulatorze
- AND, OR, XOR - argumenty jak w arytmetyce
- NOT - akumulatora
- P - parzystość
- OV - overflow
- RS0 RS1 - bank rejestrów
- AC - przeniesienie połówkowe
- CY - przeniesienie
- F0 - znacznik programowy ogólnego przeznaczenia

(hex):	D7	D6	D5	D4	D3	D2	D1	D0	
(hex): D0	CY	AC	F0	RS1	RS0	OV	—	P	PSW
	(MSB)							(LSB)	





# SpecRejestry

Adres (hex)	Oznaczenie	Nazwa
E0*	ACC	Akumulator
F0*	B	Rejestr B
D0*	PSW	Słowo stanu programu
81	SP	Wskaźnik stosu
83	DPH	bity 8–15 } wskaźnik danych bity 0–7 } DPTR
82	DPL	
80*	P0	Port 0
90*	P1	Port 1
A0*	P2	Port 2
B0*	P3	Port 3
B8*	IP	Słowo sterujące do ustawiania priorytetów przerw
A8*	IE	Słowo sterujące pracą systemu przerw
88*	TCON	Słowo sterujące i kontrolne układu czasowo-licznikowego i systemu przerw
89	TMOD	Słowo sterujące do ustawiania trybu pracy liczników w układzie czasowo-licznikowym
8C	TH0	bity 8–15 } 16-bitowy bity 0–7 } licznik T0
8A	TL0	
8D	TH1	bity 8–15 } 16-bitowy bity 0–7 } licznik T1
8B	TL1	
C8*	T2CON	Słowo sterujące licznika 2
CD	TH2	bity 8–15 } 16-bitowy bity 0–7 } licznik T2
CC	TL2	
CB	RLDH	bity 8–15 } Słowo ładowane bity 0–7 } do licznika T2
CA	RLDL	
98*	SCON	Słowo sterujące portu szeregowego
99	SBUF	Bufor portu szeregowego
97	PCON	Słowo sterujące zasilania

# A tak ładnie rozkaz przetwarzam

$S1 \longrightarrow S2 \longrightarrow S3 \longrightarrow S4 \longrightarrow S5 \longrightarrow S6$

- S1 - rozkazu pobranie,
- S2 - rozkazu dekodowanie,
- S3 - adresu efektywnego wyznaczanie,
- S4 - argumentów ładowanie,
- S5 - egzekucja,
- S6 - rezultatów pamiętanie,
- Osoby dramatu:
- Procesor: vždy - siempre!
- Pamięć: S1, S3, S4, S6.

# Faza S3 = tryby adresowania

- Rejestrowy (bezpośredni):
  - argumenty i wyniki w rejestrach procesora
  - szybki: S3 - wskazanie rejestru, S4 - szczytkowe
- Rejestrowy (pośredni):
  - zawartość rejestru adresuje pamięć (układy I/O)
  - wolniejszy niż poprzedni
- Bezpośredni:
  - adres po rozkazie pokazuje gdzie argumenty
  - S3 - proste w realizacji, S4 - długo trwa
- Natychmiastowy:
  - argumenty razem z rozkazem - błyskawica
  - S3, S4 nie istnieje, czasem i S6 brak, argumenty szczytkowe
- Bazowo-indeksowy:
  - nie jest do końca nowy, ale jest złożony!



# To umiem (1)

Mnemonic	Operacja	Kod szesnastkowy (hex)	Bajty/ cykle
<b>Przesłania</b>			
MOV A,Rr	$A \leftarrow Rr$	E8-EF	1/1
MOV A,ad	$A \leftarrow (ad)$	E5	2/1
MOV A,@Ri	$A \leftarrow (Ri)$	E6,E7	1/1
MOV A,#n	$A \leftarrow n$	74	2/1
MOV Rr,A	$Rr \leftarrow A$	F8-FF	1/1
MOV Rr,ad	$Rr \leftarrow (ad)$	A8-AF	2/2
MOV Rr,#n	$Rr \leftarrow n$	78-7F	2/1
MOV ad,A	$(ad) \leftarrow A$	F5	2/1
MOV ad,Rr	$(ad) \leftarrow Rr$	88-8F	2/2
MOV ad1,ad2	$(ad1) \leftarrow (ad2)$	85	3/2
MOV ad,@Ri	$(ad) \leftarrow (Ri)$	86,87	2/2
MOV ad,#n	$(ad) \leftarrow n$	75	3/2
MOV @Ri,A	$(Ri) \leftarrow A$	F6,F7	1/1
MOV @Ri,ad	$(Ri) \leftarrow (ad)$	A6,A7	2/2
MOV @Ri,#n	$(Ri) \leftarrow n$	76,77	2/1
MOV DPTR,#nn	$DPTR \leftarrow nn$	90	3/2
XCH A,Rr	$A \leftrightarrow Rr$	C8-CF	1/1
XCH A,ad	$A \leftrightarrow (ad)$	C5	2/1
XCH A,@Ri	$A \leftrightarrow (Ri)$	C6,C7	1/1
XCHD A,@Ri	$A_{3-0} \leftrightarrow (Ri)_{3-0}$	D6,D7	1/1
MOVX A,@Ri	$A \leftarrow (Ri)$	E2,E3	1/2
MOVX @Ri,A	$(Ri) \leftarrow A$	F2,F3	1/2
MOVX A,@DPTR	$A \leftarrow (DPTR)$	E0	1/2
MOVX @DPTR,A	$(DPTR) \leftarrow A$	F0	1/2
MOVC A,@A+DPTR	$A \leftarrow (A + DPTR)$	93	1/2
MOVC A,@A+PC	$A \leftarrow (A + PC)$	83	1/2



# To umiem (2)

---

## Operacje arytmetyczno-logiczne

---

<b>ADD A,Rr</b>	$A \leftarrow A + Rr$	<b>28–2F</b>	<b>1/1</b>
<b>ADD A,ad</b>	$A \leftarrow A + (ad)$	<b>25</b>	<b>2/1</b>
<b>ADD A,@Ri</b>	$A \leftarrow A + (Ri)$	<b>26,27</b>	<b>1/1</b>
<b>ADD A,#n</b>	$A \leftarrow A + n$	<b>24</b>	<b>2/1</b>
<b>ADDC A,Rr</b>	$A \leftarrow A + Rr + CY$	<b>38–3F</b>	<b>1/1</b>
<b>ADDC A,ad</b>	$A \leftarrow A + (ad) + CY$	<b>35</b>	<b>2/1</b>
<b>ADDC A,@Ri</b>	$A \leftarrow A + (Ri) + CY$	<b>36,37</b>	<b>1/1</b>
<b>ADDC A,#n</b>	$A \leftarrow A + n + CY$	<b>34</b>	<b>2/1</b>
<b>SUBB A,Rr</b>	$A \leftarrow A - Rr - CY$	<b>98–9F</b>	<b>1/1</b>

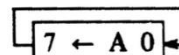
---



# To umiem (3)

Mnemonik	Operacja	Kod szesnastkowy (hex)	Bajty/cykle
SUBB A,ad	$A \leftarrow A - (ad) - CY$	95	2/1
SUBB A,@Ri	$A \leftarrow A - (Ri) - CY$	96,97	1/1
SUBB A,#n	$A \leftarrow 3A - n - CY$	94	2/1
INC A	$A \leftarrow A + 1$	04	1/1
INC Rr	$Rr \leftarrow Rr + 1$	08-0F	1/1
INC ad	$(ad) \leftarrow (ad) + 1$	05	2/1
INC @Ri	$(Ri) \leftarrow (Ri) + 1$	06,07	1/1
INC DPTR	$DPTR \leftarrow DPTR + 1$	A3	1/2
DEC A	$A \leftarrow A - 1$	14	1/1
DEC Rr	$Rr \leftarrow Rr - 1$	18-1F	1/1
DEC ad	$(ad) \leftarrow (ad) - 1$	15	2/1
DEC @Ri	$(Ri) \leftarrow (Ri) - 1$	16,17	1/1
MUL AB	$B.A \leftarrow A \times B$	A4	1/4
DIV AB	$A \leftarrow [A/B]$	84	1/4
DA A	Korekcja dzies. A	D4	1/1
ANL A,Rr	$A \leftarrow A \wedge Rr$	58-5F	1/1
ANL A,ad	$A \leftarrow A \wedge (ad)$	55	2/1
ANL A,@Ri	$A \leftarrow A \wedge (Ri)$	56,57	1/1
ANL A,#n	$A \leftarrow A \wedge n$	54	2/1
ANL ad,A	$(ad) \leftarrow (ad) \wedge A$	52	2/1
ANL ad,#n	$(ad) \leftarrow (ad) \wedge n$	53	3/2
ORL A,Rr	$A \leftarrow A \vee Rr$	48-4F	1/1
ORL A,ad	$A \leftarrow A \vee (ad)$	45	2/1
ORL A,@Ri	$A \leftarrow A \vee (Ri)$	46,47	1/1
ORL A,#n	$A \leftarrow A \vee n$	44	2/1
ORL ad,A	$(ad) \leftarrow (ad) \vee A$	42	2/1
ORL ad,#n	$(ad) \leftarrow (ad) \vee n$	43	3/2
XRL A,Rr	$A \leftarrow A \oplus Rr$	68-6F	1/1
XRL A,ad	$A \leftarrow A \oplus (ad)$	65	2/1
XRL A,@Ri	$A \leftarrow A \oplus (Ri)$	66,67	1/1
XRL A,#n	$A \leftarrow A \oplus n$	64	2/1
XRL ad,A	$(ad) \leftarrow (ad) \oplus A$	62	2/1
XRL ad,#n	$(ad) \leftarrow (ad) \oplus n$	63	3/2
CLR A	$A \leftarrow 0$	E4	1/1
CPL A	$A \leftarrow \overline{A}$	F4	1/1
SWAP A	$A_{3-0} \leftrightarrow A_{7-4}$	C4	1/1

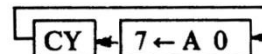
RL A



23

1/1

RLC A



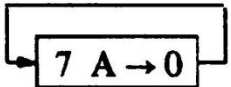
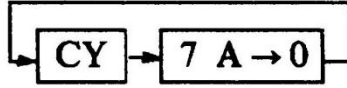
33

1/1





# To umiem (4)

Mnemonik	Operacja	Kod szesnastkowy (hex)	Bajty/ cykle
RR A		03	1/1
RRC A		13	1/1
Operacje na bitach			
CLR C	$CY \leftarrow 0$	C3	1/1
CLR bit	$(bit) \leftarrow 0$	C2	2/1
SETB C	$CY \leftarrow 1$	D3	1/1
SETB bit	$(bit) \leftarrow 1$	D2	2/1
CPL C	$CY \leftarrow \overline{CY}$	B3	1/1
CPL bit	$(bit) \leftarrow \overline{(bit)}$	B2	2/1
ANL C,bit	$CY \leftarrow CY \wedge (bit)$	82	2/2
ANL C,/bit	$CY \leftarrow CY \wedge \overline{(bit)}$	B0	2/2
ORL C,bit	$CY \leftarrow CY \vee (bit)$	72	2/2
ORL C,/bit	$CY \leftarrow CY \vee \overline{(bit)}$	A0	2/2
MOV C,bit	$CY \leftarrow (bit)$	A2	2/1
MOV bit,C	$(bit) \leftarrow CY$	92	2/2



# To umie procesor z siwą bródką (5)

## Skoki i rozkazy sterujące

<b>AJMP</b> adr11	$PC_{10-0} \leftarrow \text{adr11}$	01,21 41,61 81,A1 C1,E1	2/2
<b>LJMP</b> adr16	$PC \leftarrow \text{adr16}$	02	3/2
<b>SJMP</b> d	$PC \leftarrow PC + d$	80	2/2
<b>JMP</b> @A + DPTR	$PC \leftarrow A + \text{DPTR}$	73	1/2
<b>JC</b> d	Gdy CY = 1 to $PC \leftarrow PC + d$	40	2/2
<b>JNC</b> d	Gdy CY = 0 to $PC \leftarrow PC + d$	50	2/2
<b>JZ</b> d	Gdy A = 0 to $PC \leftarrow PC + d$	60	2/2
<b>JNZ</b> d	Gdy A $\neq$ 0 to $PC \leftarrow PC + d$	70	2/2
<b>JB</b> bit,d	Gdy (bit) = 1 to $PC \leftarrow PC + d$	20	3/2
<b>JNB</b> bit,d	Gdy (bit) = 0 to $PC \leftarrow PC + d$	30	3/2



# To umie procesor z siwą bródką (6)

Mnemonik	Operacja	Kod szesnastkowy (hex)	Bajty/ cykle
JBC bit,d	Gdy (bit) = 1 to $PC \leftarrow PC + d$ (bit) $\leftarrow$ 0	10	3/2
CJNE A,ad,d	Gdy A = (ad) to $PC \leftarrow PC + d$	B5	3/2
CJNE A,#n,d	Gdy A = n to $PC \leftarrow PC + d$	B4	3/2
CJNE Rr,#n,d	Gdy Rr = n to $PC \leftarrow PC + d$	B8–BF	3/2
CJNE @Ri,#n,d	Gdy (Ri) = n to $PC \leftarrow PC + d$	B6,B7	3/2
DJNZ Rr,d	$Rr \leftarrow Rr - 1$ i gdy $Rr \neq 0$ to $PC \leftarrow PC + d$	D8–DF	2/2
DJNZ ad,d	(ad) $\leftarrow$ (ad) – 1 i gdy (ad) $\neq 0$ to $PC \leftarrow PC + d$	D5	3/2
NOP	Nic nie rób	00	1/1



# To umie procesor z siwą bródką (7)

Podprogramy i operacje na stosie			
ACALL adr11	SP $\leftarrow$ SP + 1	11,31	2/2
	(SP) $\leftarrow$ PC <sub>7-0</sub>	51,71	
	SP $\leftarrow$ SP + 1	91,B1	
	(SP) $\leftarrow$ PC <sub>15-8</sub>	D1,F1	
	PC <sub>10-0</sub> $\leftarrow$ adr11		
LCALL adr16	SP $\leftarrow$ SP + 1	12	3/2
	(SP) $\leftarrow$ PC <sub>7-0</sub>		
	SP $\leftarrow$ SP + 1		
	(SP) $\leftarrow$ PC <sub>15-8</sub>		
	PC $\leftarrow$ adr16		
RET	PC <sub>15-8</sub> $\leftarrow$ (SP)	22	1/2
	SP $\leftarrow$ SP - 1		
	PC <sub>7-0</sub> $\leftarrow$ (SP)		
	SP $\leftarrow$ SP - 1		
RETI	PC <sub>15-8</sub> $\leftarrow$ (SP)	32	1/2
	SP $\leftarrow$ SP - 1		
	PC <sub>7-0</sub> $\leftarrow$ (SP)		
	SP $\leftarrow$ SP - 1		
PUSH ad	SP $\leftarrow$ SP + 1	C0	2/2
	(SP) $\leftarrow$ (ad)		
POP ad	(ad) $\leftarrow$ (SP)	D0	2/2
	SP $\leftarrow$ SP - 1		



# To umie procesor z siwą bródką (8)

Rozkaz	Znaczники			Rozkaz	Znaczники			Rozkaz	Znaczники		
	CY	AC	OV		CY	AC	OV		CY	AC	OV
<b>ADD</b>	↑	↑	↑	<b>RRC</b>	↑	–	–	<b>ANL C,bit</b>	↑	–	–
<b>ADDC</b>	↑	↑	↑	<b>RLC</b>	↑	–	–	<b>ANL C,/bit</b>	↑	–	–
<b>SUBB</b>	↑	↑	↑	<b>CJNE</b>	↑	–	–	<b>ORL C,bit</b>	↑	–	–
<b>MUL</b>	0	–	↑	<b>SETB C</b>	1	–	–	<b>ORL C,/bit</b>	↑	–	–
<b>DIV</b>	0	–	↑	<b>CLR C</b>	0	–	–	<b>MOV C,bit</b>	↑	–	–
<b>DA</b>	↑	–	–	<b>CPL C</b>	↑	–	–				