

Opis '51:

(na podstawie „Mikrokomputery jednokładowe rodziny MCS-51” A. Rydzewskiego)

Mikrokomputer jednokładowy 8051:

- schemat blokowy mikrokomputera – aneks 2,
- zegar max. 12 MHz (najnowsze wersje z jądrem RISC-owym do 40 MHz),
- osobne przestrzenie adresowe pamięci programu (4 KB ROM – adresowanie specjalną instrukcją MOVC) i pamięci danych (128 B RAM); oba rodzaje pamięci rozszerzalne zewnętrznie do 64 KB każda (adresowanie dodatkowej pamięci RAM specjalną instrukcją MOVX),
- 2 16-bitowe liczniki, konfigurowalne do pracy w 4 trybach, taktowane zegarem lub sygnałem zewnętrznym,
- 4 8-bitowe porty (32 sygnały niezależnie i dowolnie konfigurowalne jako we/wy), część z nich może być specjalnie wykorzystywana (np. do podłączenia zewnętrznej pamięci)
- układ obsługi portu szeregowego, pracującego w 4 trybach,
- 2 poziomy układ przerwań (3 rodzaje przerwań: z 2 linii przerwań zewnętrznych, 2 generowane przez liczniki oraz 1 generowane przez port szeregowy),
- lista rozkazów obejmuje instrukcje:
 - przesłań,
 - arytmetyczne,
 - logiczne na bajtach,
 - logiczne na bitach,
 - sterowania przebiegiem programu (w tym skoki (bez)warunkowe i wywołania procedur);
- tabela listy instrukcji – aneks 3 i aneks 6,

Bardziej szczegółowe informacje:

- mapa pamięci RAM – aneks 4,
- Rejestry:
 - ACC (A) – akumulator,
 - B – rejestr ogólnego przeznaczenia,
 - PSW – Program Status Word:
 - P – parzystość bitów akumulatora (P=0, to ilość parzysta),
 - nieużywany,
 - OV – overflow – wystąpienie nadmiaru przy 8-bitowych operacjach na liczbach U2,
 - RS0 i RS1 – flagi wyboru banku rejestrów,
 - F0 – flaga użytkownika (dostępna tylko „explicite” programowo),
 - AC – auxiliary carry – używana m.in. przy korekcji dziesiętnej (DA A),
 - CY – carry – przeniesienie dla operacji na liczbach bez znaku oraz „akumulator” dla operacji bitowych,
 - RS0, RS1 – wybór banku rejestrów R0-R7 – patrz niżej,
 - PC – 16-bitów
 - SP – 16-bitów: przed każdym zapisem na stos INC SP, po każdym odczycie DEC SP,
 - 4 banki 8 rejestrów R0-R7, bank wybierany 2 bitami RS (Register Bank Switch), zawartymi w PSW;
 - rejestry R0 i R1 są wykorzystywane w większej ilości instrukcji do adresowania niż pozostałe R2-R7,

- SFR – zestaw Special Functions Registers – dostęp tylko adresowaniem bezpośrednim.
- adresowanie bitowe – niektóre SFR oraz bajty 20h÷ 2Fh – mapa adresów bitów – aneks 1,
- '52 ma na adresach SFR dodatkowe 128B pamięci adresowanych rejestrami R0/R1 (bo dostęp do SFR jest tylko adresowaniem bezpośrednim),
- niektóre adresy bitów oraz adresy rejestrów z grupy SFR predefiniowane w assemblerach jako stałe symboliczne,
- po RESET'ie program wykonuje się od adresu 0, natomiast po otrzymaniu przerwania {~INT0, T0, ~INT1, T1, SerialPort, T2('52)} od adresu {3h, Bh, 13h, 1Bh, 23h, 2Bh }
- 4 porty (bramy) po 8 linii: 32 linie konfigurowalne niezależnie (jeśli ma być wejściowa, to należy w ten bit wpisać „1” i odbierać z niego faktyczny stan wejścia, jeśli wyjściowa, to wpisać wartość, jaka ma się pojawić na wyjściu). Tylko P1 jest wykorzystywany jedynie jako we/wy, reszta może służyć również innym celom, np.:
 - P3.0-P3.7={Port szeregowy (RXD, TXD),
 - linie zewnętrznych przerw (INT0, INT1),
 - linie zewnętrznego taktowania liczników (T0, T1),
 - obsługa zewnętrznej pamięci RAM (WR, RD)}.

Instrukcje wymagające jedynie odczytu lub jedynie zapisu danych do portu, działają na końcówkach (pinach) układu, natomiast instrukcje modyfikujące jego zawartość (np. INC P0) wymagające akcji read-modify-write, odczytują informację z buforowego rejestru wyjściowego. Zapis zawsze odbywa się do buforowego rejestru wyjściowego, natomiast przesłanie bufora na końcówki (piny) jest opóźniony do następnej instrukcji.

- 5 rodzajów trybu adresowania:
 - rejestrowe – R0 ... R7 z wybranego banku rejestrów, ACC, B, CY (bit), DPTR,
 - bezpośrednie – wewnętrzna pamięć RAM oraz obszar rejestrów specjalnych,
 - pośrednie rejestrowe – wewnętrzna pamięć RAM (@R0, @R1, SP) oraz zewnętrzna pamięć danych (@R0, @R1, @DPTR),
 - natychmiastowe – pamięć programu,
 - indeksowe – pamięć programu: @A+DPTR, @A+PC.

Aneks 1 – Mapa adresów bitów

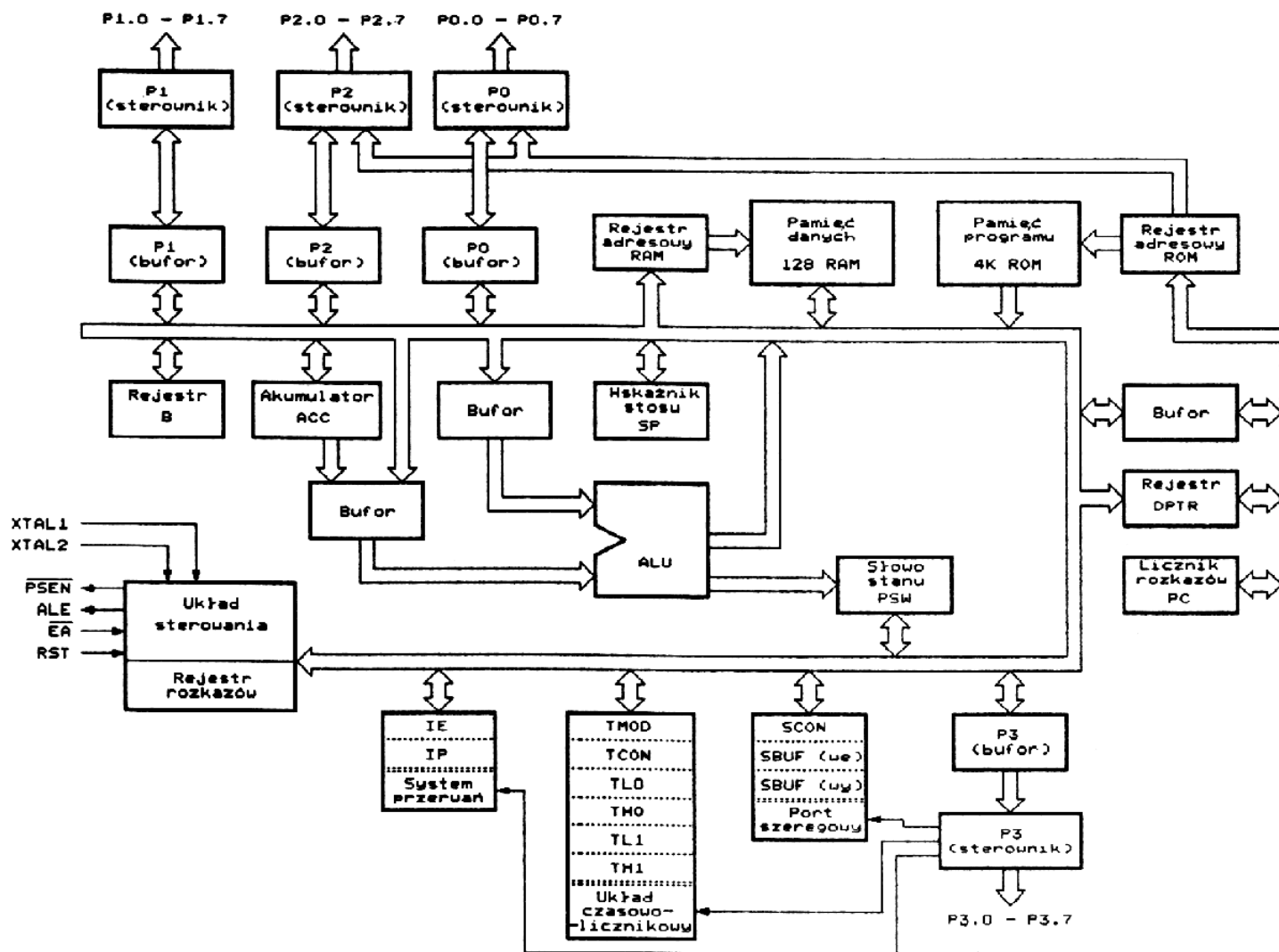
Adres bajtu	Adres bitu (hex)							
127 (7FH)								
48 (30H)								
47 (2FH)	7F	7E	7D	7C	7B	7A	79	78
46 (2EH)	77	76	75	74	73	72	71	70
45 (2DH)	6F	6E	6D	6C	6B	6A	69	68
44 (2CH)	67	66	65	64	63	62	61	60
43 (2BH)	5F	5E	5D	5C	5B	5A	59	58
42 (2AH)	57	56	55	54	53	52	51	50
41 (29H)	4F	4E	4D	4C	4B	4A	49	48
40 (28H)	47	46	45	44	43	42	41	40
39 (27H)	3F	3E	3D	3C	3B	3A	39	38
38 (26H)	37	36	35	34	33	32	31	30
37 (25H)	2F	2E	2D	2C	2B	2A	29	28
36 (24H)	27	26	25	24	23	22	21	20
35 (23H)	1F	1E	1D	1C	1B	1A	19	18
34 (22H)	17	16	15	14	13	12	11	10
33 (21H)	0F	0E	0D	0C	0B	0A	09	08
32 (20H)	07	06	05	04	03	02	01	00
31 (1FH)	Rejestry robocze zbiór 3							
24 (18H)	Rejestry robocze zbiór 2							
23 (17H)	Rejestry robocze zbiór 1							
16 (10H)	Rejestry robocze zbiór 1							
15 (0FH)	Rejestry robocze zbiór 1							
8 (08H)	Rejestry robocze zbiór 0							
7 (07H)	Rejestry robocze zbiór 0							
0 (00H)	Rejestry robocze zbiór 0							

Pamięć użytkowa

Adres bajtu	Adres bitu (hex)								Symbol rejestru
240 (F0H)	F7	F6	F5	F4	F3	F2	F1	F0	B
224 (E0H)	E7	E6	E5	E4	E3	E2	E1	E0	ACC
208 (D0H)	CY	AC	F0	RS1	RS0	OV	P		PSW
200 (C8H)	D7	D6	D5	D4	D3	D2	D1	D0	T2CON
	TF2	EXF2	RCLK	EXEN2	ICLK	TR2	C/T2	CP/RL2	
184 (B8H)	PT2	PS	PT1	PX1	PT0	PX0	--		IP
176 (B0H)	B7	B6	B5	B4	B3	B2	B1	B0	P3
168 (A8H)	EA	ET2	ES	ET1	EX1	ET0	EX0	AF	IE
160 (A0H)	A7	A6	A5	A4	A3	A2	A1	A0	P2
152 (98H)	SM0	SM1	SM2	REN	TB8	R88	TI	RI	SCON
144 (90H)	9F	9E	9D	9C	9B	9A	99	98	P1
	97	96	95	94	93	92	91	90	
136 (88H)	IF1	TR1	TF0	TRO	IE1	IT1	IE0	IT0	TCON
128 (80H)	B7	B6	B5	B4	B3	B2	B1	B0	PO

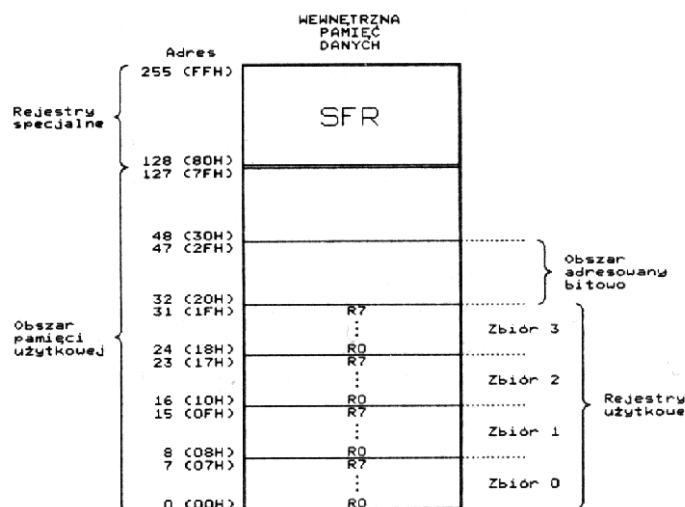
Rejestry specjalne

Aneks 2 – Schemat blokowy mikrokomputera 8051



Aneks 4 – Mapa pamięci RAM oraz zestawienie rejestrów specjalnych

Rejestry zaznaczone * osiągalne są również przez adresowanie bitowe



TABLICA 2.1. Zestawienie rejestrów specjalnych

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

Aneks 5 – Instrukcje zmieniające flagi

Instrukcja	C	Ov	AC
ADD	X	X	X
ADDC	X	X	X
SUBB	X	X	X
MUL	0	X	-
DIV	0	X	-
DA	X	-	-
RRC	X	-	-
RLC	X	-	-
SETB C	1	-	-
CLR C	0	-	-
CPL C	X	-	-
ANL C,bit	X	-	-
ANL C,/bit	X	-	-
ORL C,bit	X	-	-
ORL C,/bit	X	-	-
MOV C,bit	X	-	-
CJNE	X	-	-

Aneks 6 – Opis instrukcji / długa wersja listy instrukcji:

Lista rozkazów mikrokontrolerów rodziny '51 zawiera 111 instrukcji (49 jednobajtowych, 45 dwubajtowych i 17 trzybajtowych). W poniższym zestawieniu uwzględniono jedynie mnemoniki poszczególnych rozkazów, które są rozpoznawane przez asemblery '51 i one właśnie są najczęściej stosowane przy tworzeniu programu.

W zestawieniu stosuje się następujące oznaczenia:

- Rr - rejestry R0-R7.
- direct - obszar wewnętrznej pamięci danych (128B) i rejestry SFR.
- @Ri - adres z rejestrów indeksowych R0 lub R1.
- bit - bity pamięci danych i SFR adresowalne bezpośrednio.
- #data - stała 8-bitowa.
- #data16 - stała 16-bitowa.
- addr16 - adres 16-bitowy (obszar 64kB).
- addr11 - adres 11-bitowy (w stronie 2kB).
- rel - adres względny -128...+127.
- /bit - negacja bitu.

Rozkazy mikrokontrolerów rodziny '51 mogą zostać pogrupowane w następujący sposób:

- Operacje arytmetyczne.
- Operacje logiczne.
- Operacje logiczne na bitach.
- Przesyłanie danych.
- Skoki.

Operacje arytmetyczne

Mnem	Funkcja	Opis	Tryby adr.	Zmien. flagi
ADD	dodawanie	$A=A+data$	A,Rr ; A,direct ; A,@Ri ; A,#data	C, AC, Ov
ADDC	dodawanie z przeniesieniem	$A=A+data+C$	A,Rr ; A,direct ; A,@Ri ; A,#data	C, AC, Ov
SUBB	odejmowanie z pożyczką	$A=A-dana-C$	A,Rr ; A,direct ; A,@Ri ; A,#data	C, AC, Ov
INC	inkrementacja	argument=argument+1	A ; Rr ; direct ; @Ri ; DPTR	
DEC	dekrementacja	argument=argument-1	A ; Rr ; direct ; @Ri	
MUL	mnożenie liczb 8-bitowych, wynik 16-bitowy	$A*B$, A - młodsza część wyniku, B - starsza	AB	Ov, C=0
DIV	dzielenie liczb 8-bitowych, 8-bitowy wynik i 8-bitowa reszta	A/B , A – wynik, B – reszta	AB	Ov, C=0
DA	poprawka dziesiętna	poprawka dziesiętna liczby 8-bitowej, przekształcając ją do postaci 2 cyfr w kodzie BCD	A	C

Operacje logiczne na bitach

CLR	zerowanie	zeruje podany bit	C ; bit	
SETB	ustawienie bitu	przypisuje bitowi wartość 1	C ; bit	
CPL	negacja	neguje podany bit	C ; bit	
ANL	iloczyn logiczny AND	wykonuje iloczyn logiczny dwóch bitów, wynik zapisuje w pierwszym	C,bit ; C,/bit	
ORL	suma logiczna OR	wykonuje sumę logiczną dwóch bitów, wynik zapisuje w pierwszym	C,bit ; C,/bit	
MOV	kopiuj	wpisuje do pierwszego bitu wartość zapisaną w drugim	C,bit ; bit,C	

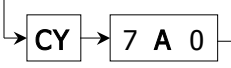
Skoki

ACALL	wywołanie podprogramu w stronie 2kB	zapisuje na stos PC i wykonuje skok pod addr11	addr11	
LCALL	wywołanie podprogramu	zapisuje na stos PC i wykonuje skok pod addr16	addr16	
RET	powrót z procedury	pobiera ze stosu PC i przekazuje sterowanie pod ten adres		
RETI	powrót z procedury obsługi przerwania	pobiera ze stosu PC i przekazuje sterowanie pod ten adres		
LJMP	rozkaz skoku w obszarze 64kB	wpisuje do PC podany adres	addr16	
AJMP	rozkaz skoku w stronie 2kB	zastępuje 11 młodszych bitów PC wartością podaną w wywołaniu	addr11	
SJMP	rozkaz skoku w obszarze 256B	następuje zmiana zawartości PC o wartość 'rel' w kodzie U2	rel	
JMP	rozkaz skoku w trybie pośrednim	przepisuje do PC sumę zawartości DPTR i rejestru bazowego	@A+DPTR	
JZ	skok warunkowy	wykonuje skok, jeżeli ACC=0	rel	
JNZ	skok warunkowy	wykonuje skok, jeżeli ACC≠0	rel	
JC	skok warunkowy	wykonuje skok, jeżeli CY=1	rel	
JNC	skok warunkowy	wykonuje skok, jeżeli CY=0	rel	
JB	skok warunkowy	wykonuje skok, jeżeli dany bit jest ustawiony	bit,rel	
JNB	skok warunkowy	wykonuje skok, jeżeli dany bit jest wyzerowany	bit,rel	
JBC	skok warunkowy	gdy dany bit jest ustawiony, to nastąpi skok i wyzerowanie bitu	bit,rel	
CJNE	skok warunkowy, zależny od wyniku porównania	następuje porównanie operandów, jeżeli są różne, to następuje skok	A,direct,rel ; A,#data,rel ; Rr,#data,rel ; @Ri,#data,rel	C=1 jeśli 1. operand ^ 2. operand
DJNZ	skok warunkowy, zależny od wyniku dekrementacji	następuje dekrementacja zawartości operandu, jeżeli zawartość ≠ 0, to nastąpi skok	Rr,rel ; direct,rel	

Inne

NOP	nic nie rób			
------------	-------------	--	--	--

Operacje logiczne

ANL	iloczyn logiczny AND	logiczny iloczyn na odpowiednich bitach argumentów, wynik zapisuje do pierwszego argumentu	A,Rr ; A,direct ; A,@Ri ; A,#data ; direct,A ; direct,#data	
ORL	suma logiczna OR	suma logiczna odpowiednich bitów argumentów, wynik zapisuje do pierwszego argumentu	A,Rr ; A,direct ; A,@Ri ; A,#data ; direct,A ; direct,#data	
XRL	suma mod 2 (XOR)	sumowanie mod 2 odpowiednich bitów argumentu, wynik zapisuje do pierwszego z nich	A,Rr ; A,direct ; A,@Ri ; A,#data ; direct,A ; direct,#data	
CLR	zerowanie (bajtowe)	zeruje wszystkie bity argumentu	A	
CPL	negowanie (bajtowe)	neguje wszystkie bity argumentu	A	
RL	obrót w lewo	przesuwa w lewo zawartość akumulatora	A	
RR	obrót w prawo	przesuwa w prawo zawartość akumulatora	A	
RLC	obrót w lewo przez C	c	A	
RRC	obrót w prawo przez C		A	
SWAP	zamiana 4 bitów bajtu	zamienia miejscami 4 starsze i 4 młodsze bity bajtu	A	

Przesyłanie danych

MOV	kopiuje	wpisuje do pierwszego operandu zawartość drugiego	A,Rr ; A,direct ; A,@Ri ; A,#data ; DPTR,#data16	direct,Rr ; direct,direct ; direct,@Ri ; direct,#data ; direct,A ;	Rr,A ; @Ri,A ; Rr,direct ; @Ri,direct ; Rr,#data ; @Ri,data ;	
MOV	kopiuje z pamięci programu	kopiuje bajt kodu (zawartości pamięci programu)			A,@A+DPTR ; A,@A+PC	
MOVX	kopiuje z/do zewnętrznej pamięci danych	kopiuje bajt z/do komórki zewnętrznej pamięci danych wskazanej rejestrem indeksowym			A,@Ri ; A,@DPTR ; @Ri,A ; @DPTR,A	
XCH	zamień	wymienia między sobą zawartość dwóch operandów			A,Rr ; A,direct ; A,@Ri	
XCHD	zamień młodsze 4 bity	wymienia między operandami 4 młodsze bity			A,@Ri	
PUSH	przesłanie bajtu na stos	zwiększa SP i kopiuje zawartość podanej komórki pod adres w SP			direct	
POP	pobranie bajtu ze stosu	kopiuje zawartość komórki spod adresu w SP do podanej i zmniejsza SP			direct	