

Kode ASCII

ASCII (American Standard Code for Information Interchange) merupakan Kode Standar Amerika untuk Pertukaran Informasi atau sebuah standar internasional dalam pengkodean huruf dan simbol seperti Unicode dan Hex tetapi ASCII lebih bersifat universal.

Tabel Code ASCII

DEC	OCT	HEX	BIN	Symbol	Description
0	0	0	0	NUL	Null char
1	1	1	1	SOH	Start of Heading
2	2	2	10	STX	Start of Text
3	3	3	11	ETX	End of Text
4	4	4	100	EOT	End of Transmission
5	5	5	101	ENQ	Enquiry
6	6	6	110	ACK	Acknowledgment
7	7	7	111	BEL	Bell
8	10	8	1000	BS	Back Space
9	11	9	1001	HT	Horizontal Tab
10	12	0A	1010	LF	Line Feed
11	13	0B	1011	VT	Vertical Tab
12	14	0C	1100	FF	Form Feed
13	15	0D	1101	CR	Carriage Return
14	16	0E	1110	SO	Shift Out / X-On
15	17	0F	1111	SI	Shift In / X-Off

16	20	10	10000	DLE	Data Line Escape
17	21	11	10001	DC1	Device Control 1 (oft. XON)
18	22	12	10010	DC2	Device Control 2
19	23	13	10011	DC3	Device Control 3 (oft. XOFF)
20	24	14	10100	DC4	Device Control 4
21	25	15	10101	NAK	Negative Acknowledgement
22	26	16	10110	SYN	Synchronous Idle
23	27	17	10111	ETB	End of Transmit Block
24	30	18	11000	CAN	Cancel
25	31	19	11001	EM	End of Medium
26	32	1A	11010	SUB	Substitute
27	33	1B	11011	ESC	Escape
28	34	1C	11100	FS	File Separator
29	35	1D	11101	GS	Group Separator

30	36	1E	11110	RS	Record Separator
31	37	1F	11111	US	Unit Separator
32	40	20	100000		Space
33	41	21	100001	!	Exclamation mark
34	42	22	100010	"	Double quotes (or speech marks)
35	43	23	100011	#	Number
36	44	24	100100	\$	Dollar
37	45	25	100101	%	Per cent sign
38	46	26	100110	&	Ampersand
39	47	27	100111	'	Single quote
40	50	28	101000	(Open parenthesis (or open bracket)
41	51	29	101001)	Close parenthesis (or close bracket)
42	52	2A	101010	*	Asterisk
43	53	2B	101011	+	Plus
44	54	2C	101100	,	Comma
45	55	2D	101101	-	Hyphen
46	56	2E	101110	.	Period, dot or full stop
47	57	2F	101111	/	Slash or divide
48	60	30	110000	0	Zero
49	61	31	110001	1	One
50	62	32	110010	2	Two
51	63	33	110011	3	Three
52	64	34	110100	4	Four
53	65	35	110101	5	Five
54	66	36	110110	6	Six
55	67	37	110111	7	Seven
56	70	38	111000	8	Eight
57	71	39	111001	9	Nine
58	72	3A	111010	:	Colon
59	73	3B	111011	;	Semicolon
60	74	3C	111100	<	Less than (or open angled bracket)
61	75	3D	111101	=	Equals

62	76	3E	111110	>	Greater than (or close angled bracket)
63	77	3F	111111	?	Question mark
64	100	40	1E+06	@	At symbol
65	101	41	1E+06	A	Uppercase A
66	102	42	1E+06	B	Uppercase B
67	103	43	1E+06	C	Uppercase C
68	104	44	1E+06	D	Uppercase D
69	105	45	1E+06	E	Uppercase E
70	106	46	1E+06	F	Uppercase F
71	107	47	1E+06	G	Uppercase G
72	110	48	1E+06	H	Uppercase H
73	111	49	1E+06	I	Uppercase I
74	112	4A	1E+06	J	Uppercase J
75	113	4B	1E+06	K	Uppercase K
76	114	4C	1E+06	L	Uppercase L
77	115	4D	1E+06	M	Uppercase M
78	116	4E	1E+06	N	Uppercase N
79	117	4F	1E+06	O	Uppercase O
80	120	50	1E+06	P	Uppercase P
81	121	51	1E+06	Q	Uppercase Q
82	122	52	1E+06	R	Uppercase R
83	123	53	1E+06	S	Uppercase S
84	124	54	1E+06	T	Uppercase T
85	125	55	1E+06	U	Uppercase U
86	126	56	1E+06	V	Uppercase V
87	127	57	1E+06	W	Uppercase W
88	130	58	1E+06	X	Uppercase X
89	131	59	1E+06	Y	Uppercase Y
90	132	5A	1E+06	Z	Uppercase Z
91	133	5B	1E+06	[Opening bracket
92	134	5C	1E+06	\	Backslash
93	135	5D	1E+06]	Closing bracket
94	136	5E	1E+06	^	Caret - circumflex
95	137	5F	1E+06	_	Underscore
96	140	60	1E+06	`	Grave accent
97	141	61	1E+06	a	Lowercase a

98	142	62	1E+06	b	Lowercase b
99	143	63	1E+06	c	Lowercase c
100	144	64	1E+06	d	Lowercase d
101	145	65	1E+06	e	Lowercase e
102	146	66	1E+06	f	Lowercase f
103	147	67	1E+06	g	Lowercase g
104	150	68	1E+06	h	Lowercase h
105	151	69	1E+06	i	Lowercase i
106	152	6A	1E+06	j	Lowercase j
107	153	6B	1E+06	k	Lowercase k
108	154	6C	1E+06	l	Lowercase l
109	155	6D	1E+06	m	Lowercase m
110	156	6E	1E+06	n	Lowercase n
111	157	6F	1E+06	o	Lowercase o
112	160	70	1E+06	p	Lowercase p
113	161	71	1E+06	q	Lowercase q
114	162	72	1E+06	r	Lowercase r
115	163	73	1E+06	s	Lowercase s
116	164	74	1E+06	t	Lowercase t
117	165	75	1E+06	u	Lowercase u
118	166	76	1E+06	v	Lowercase v
119	167	77	1E+06	w	Lowercase w
120	170	78	1E+06	x	Lowercase x
121	171	79	1E+06	y	Lowercase y
122	172	7A	1E+06	z	Lowercase z
123	173	7B	1E+06	{	Opening brace
124	174	7C	1E+06		Vertical bar
125	175	7D	1E+06	}	Closing brace
126	176	7E	1E+06	~	Equivalency sign - tilde
127	177	7F	1E+06		Delete
128	200	80	1E+07	€	Euro sign
129	201	81	1E+07		
130	202	82	1E+07	,	Single low-9 quotation mark
131	203	83	1E+07	<i>f</i>	Latin small letter f with hook
132	204	84	1E+07	„	Double low-9 quotation mark

133	205	85	1E+07	...	Horizontal ellipsis
134	206	86	1E+07	†	Dagger
135	207	87	1E+07	‡	Double dagger
136	210	88	1E+07	^	Modifier letter circumflex accent
137	211	89	1E+07	‰	Per mille sign
138	212	8A	1E+07	Š	Latin capital letter S with caron
139	213	8B	1E+07	‹	Single left-pointing angle quotation
140	214	8C	1E+07	Œ	Latin capital ligature OE
141	215	8D	1E+07		
142	216	8E	1E+07	Ž	Latin capital letter Z with caron
143	217	8F	1E+07		
144	220	90	1E+07		
145	221	91	1E+07	‘	Left single quotation mark
146	222	92	1E+07	’	Right single quotation mark
147	223	93	1E+07	“	Left double quotation mark
148	224	94	1E+07	”	Right double quotation mark
149	225	95	1E+07	•	Bullet
150	226	96	1E+07	—	En dash
151	227	97	1E+07	—	Em dash
152	230	98	1E+07	~	Small tilde
153	231	99	1E+07	™	Trade mark sign
154	232	9A	1E+07	š	Latin small letter S with caron
155	233	9B	1E+07	›	Single right-pointing angle quotation mark
156	234	9C	1E+07	œ	Latin small ligature oe
157	235	9D	1E+07		
158	236	9E	1E+07	ž	Latin small letter z with caron

159	237	9F	1E+07	ÿ	Latin capital letter Y with diaeresis
160	240	A0	1E+07		Non-breaking space
161	241	A1	1E+07	¡	Inverted exclamation mark
162	242	A2	1E+07	¢	Cent sign
163	243	A3	1E+07	£	Pound sign
164	244	A4	1E+07	¤	Currency sign
165	245	A5	1E+07	¥	Yen sign
166	246	A6	1E+07		Pipe, Broken vertical bar
167	247	A7	1E+07	§	Section sign
168	250	A8	1E+07	..	Spacing diaeresis - umlaut
169	251	A9	1E+07	©	Copyright sign
170	252	AA	1E+07	ª	Feminine ordinal indicator
171	253	AB	1E+07	«	Left double angle quotes
172	254	AC	1E+07	¬	Not sign
173	255	AD	1E+07		Soft hyphen
174	256	AE	1E+07	®	Registered trade mark sign
175	257	AF	1E+07	–	Spacing macron - overline
176	260	B0	1E+07	°	Degree sign
177	261	B1	1E+07	±	Plus-or-minus sign
178	262	B2	1E+07	²	Superscript two - squared
179	263	B3	1E+07	³	Superscript three - cubed
180	264	B4	1E+07	´	Acute accent - spacing acute
181	265	B5	1E+07	µ	Micro sign
182	266	B6	1E+07	¶	Pilcrow sign - paragraph sign
183	267	B7	1E+07	·	Middle dot - Georgian comma
184	270	B8	1E+07	¸	Spacing cedilla
185	271	B9	1E+07	¹	Superscript one

186	272	BA	1E+07	◦	Masculine ordinal indicator
187	273	BB	1E+07	»	Right double angle quotes
188	274	BC	1E+07	¼	Fraction one quarter
189	275	BD	1E+07	½	Fraction one half
190	276	BE	1E+07	¾	Fraction three quarters
191	277	BF	1E+07	¿	Inverted question mark
192	300	C0	1E+07	À	Latin capital letter A with grave
193	301	C1	1E+07	Á	Latin capital letter A with acute
194	302	C2	1E+07	Â	Latin capital letter A with circumflex
195	303	C3	1E+07	Ã	Latin capital letter A with tilde
196	304	C4	1E+07	Ä	Latin capital letter A with diaeresis
197	305	C5	1E+07	Å	Latin capital letter A with ring above
198	306	C6	1E+07	Æ	Latin capital letter AE
199	307	C7	1E+07	Ç	Latin capital letter C with cedilla
200	310	C8	1E+07	È	Latin capital letter E with grave
201	311	C9	1E+07	É	Latin capital letter E with acute
202	312	CA	1E+07	Ê	Latin capital letter E with circumflex
203	313	CB	1E+07	Ë	Latin capital letter E with diaeresis
204	314	CC	1E+07	Ì	Latin capital letter I with grave
205	315	CD	1E+07	Í	Latin capital letter I with acute
206	316	CE	1E+07	Î	Latin capital letter I with circumflex

207	317	CF	1E+07	Ï	Latin capital letter I with diaeresis
208	320	D0	1E+07	Ð	Latin capital letter ETH
209	321	D1	1E+07	Ñ	Latin capital letter N with tilde
210	322	D2	1E+07	Ò	Latin capital letter O with grave
211	323	D3	1E+07	Ó	Latin capital letter O with acute
212	324	D4	1E+07	Ô	Latin capital letter O with circumflex
213	325	D5	1E+07	Õ	Latin capital letter O with tilde
214	326	D6	1E+07	Ö	Latin capital letter O with diaeresis
215	327	D7	1E+07	×	Multiplication sign
216	330	D8	1E+07	Ø	Latin capital letter O with slash
217	331	D9	1E+07	Ù	Latin capital letter U with grave
218	332	DA	1E+07	Ú	Latin capital letter U with acute
219	333	DB	1E+07	Û	Latin capital letter U with circumflex
220	334	DC	1E+07	Ü	Latin capital letter U with diaeresis
221	335	DD	1E+07	Ý	Latin capital letter Y with acute
222	336	DE	1E+07	Þ	Latin capital letter THORN
223	337	DF	1E+07	ß	Latin small letter sharp s - ess-zed
224	340	E0	1E+07	à	Latin small letter a with grave
225	341	E1	1E+07	á	Latin small letter a with acute
226	342	E2	1E+07	â	Latin small letter a with circumflex

227	343	E3	1E+07	ã	Latin small letter a with tilde
228	344	E4	1E+07	ä	Latin small letter a with diaeresis
229	345	E5	1E+07	å	Latin small letter a with ring above
230	346	E6	1E+07	æ	Latin small letter ae
231	347	E7	1E+07	ç	Latin small letter c with cedilla
232	350	E8	1E+07	è	Latin small letter e with grave
233	351	E9	1E+07	é	Latin small letter e with acute
234	352	EA	1E+07	ê	Latin small letter e with circumflex
235	353	EB	1E+07	ë	Latin small letter e with diaeresis
236	354	EC	1E+07	ì	Latin small letter i with grave
237	355	ED	1E+07	í	Latin small letter i with acute
238	356	EE	1E+07	î	Latin small letter i with circumflex
239	357	EF	1E+07	ï	Latin small letter i with diaeresis
240	360	F0	1E+07	ð	Latin small letter eth
241	361	F1	1E+07	ñ	Latin small letter n with tilde
242	362	F2	1E+07	ò	Latin small letter o with grave
243	363	F3	1E+07	ó	Latin small letter o with acute
244	364	F4	1E+07	ô	Latin small letter o with circumflex
245	365	F5	1E+07	õ	Latin small letter o with tilde
246	366	F6	1E+07	ö	Latin small letter o with diaeresis

247	367	F7	1E+07	÷	Division sign
248	370	F8	1E+07	ø	Latin small letter o with slash
249	371	F9	1E+07	ù	Latin small letter u with grave
250	372	FA	1E+07	ú	Latin small letter u with acute
251	373	FB	1E+07	û	Latin small letter u with circumflex
252	374	FC	1E+07	ü	Latin small letter u with diaeresis
253	375	FD	1E+07	ý	Latin small letter y with acute
254	376	FE	1E+07	þ	Latin small letter thorn
255	377	FF	1E+07	ÿ	Latin small letter y with diaeresis

Perintah –Perintah Dibahasa assembly Interl x86

1. ACALL (Absolute Call)

ACALL berfungsi untuk memanggil sub rutin program

2. ADD (Add Immediate Data)

ADD berfungsi untuk menambah 8 bit data langsung ke dalam isi akumulator dan menyimpan hasilnya pada akumulator.

3. ADDC (Add Carry Plus Immediate Data to Accumulator)

ADDC berfungsi untuk menambahkan isi carry flag (0 atau 1) ke dalam isi akumulator. Data langsung 8 bit ditambahkan ke akumulator.

4. AJMP (Absolute Jump)

AJMP adalah perintah jump mutlak. Jump dalam 2 KB dimulai dari alamat yang mengikuti perintah AJMP. AJMP berfungsi untuk mentransfer kendali program ke lokasi dimana alamat dikalkulasi dengan cara yang sama dengan perintah ACALL. Konter program ditambahkan dua kali dimana perintah AJMP adalah perintah 2-byte. Konter program di-load dengan a10 – a0 11 bits, untuk membentuk alamat tujuan 16-bit.

5. ANL (logical AND memori ke akumulator)

ANL berfungsi untuk mengAND-kan isi alamat data dengan isi akumulator.

6. CJNE (Compare Indirect Address to Immediate Data)

CJNE berfungsi untuk membandingkan data langsung dengan lokasi memori yang dialamati oleh register R atau Akumulator A. apabila tidak sama maka instruksi akan menuju ke alamat kode.

Format : CJNE R,#data,Alamat kode.

7. CLR (Clear Accumulator)

CLR berfungsi untuk mereset data akumulator menjadi 00H.

Format : CLR A

8. CPL (Complement Accumulator)

CPL berfungsi untuk mengkomplemen isi akumulator.

9. DA (Decimal Adjust Accumulator)

DA berfungsi untuk mengatur isi akumulator ke padanan BCD, setelah penambahan dua angka BCD.

10. DEC (Decrement Indirect Address)

DEC berfungsi untuk mengurangi isi lokasi memori yang ditunjukkan oleh register R dengan 1, dan hasilnya disimpan pada lokasi tersebut.

11. DIV (Divide Accumulator by B)

DIV berfungsi untuk membagi isi akumulator dengan isi register B. Akumulator berisi hasil bagi, register B berisi sisa pembagian.

12. DJNZ (Decrement Register And Jump If Not Zero)

DJNZ berfungsi untuk mengurangi nilai register dengan 1 dan jika hasilnya sudah 0 maka instruksi selanjutnya akan dieksekusi. Jika belum 0 akan menuju ke alamat kode.

13. INC (Increment Indirect Address)

INC berfungsi untuk menambahkan isi memori dengan 1 dan menyimpannya pada alamat tersebut.

14. JB (Jump if Bit is Set)

JB berfungsi untuk membaca data per satu bit, jika data tersebut adalah 1 maka akan menuju ke alamat kode dan jika 0 tidak akan menuju ke alamat kode.

15. JBC (Jump if Bit Set and Clear Bit)

Bit JBC, berfungsi sebagai perintah rel menguji yang terspesifikasikan secara bit. Jika bit di-set, maka Jump dilakukan ke alamat relatif dan yang terspesifikasi secara bit di dalam perintah dibersihkan. Segmen program berikut menguji bit yang kurang signifikan (LSB: Least Significant Byte), dan jika ditemukan bahwa ia telah di-set, program melompat ke READ lokasi. JBC juga berfungsi membersihkan LSB dari akumulator.

16. JC (Jump if Carry is Set)

Instruksi JC berfungsi untuk menguji isi carry flag. Jika berisi 1, eksekusi menuju ke alamat kode, jika berisi 0, instruksi selanjutnya yang akan dieksekusi.

17. JMP (Jump to sum of Accumulator and Data Pointer)

Instruksi JMP berfungsi untuk memerintahkan loncat ke suatu alamat kode tertentu.

Format : JMP alamat kode.

18. JNB (Jump if Bit is Not Set)

Instruksi JNB berfungsi untuk membaca data per satu bit, jika data tersebut adalah 0 maka akan menuju ke alamat kode dan jika 1 tidak akan menuju ke alamat kode.

Format : JNB alamat bit, alamat kode.

19. JNC (Jump if Carry Not Set)

JNC berfungsi untuk menguji bit Carry, dan jika tidak di-set, maka sebuah lompatan akan dilakukan ke alamat relatif yang telah ditentukan.

20. JNZ (Jump if Accumulator Not Zero)

JNZ adalah mnemonik untuk instruksi jump if not zero (lompat jika tidak nol). Dalam hal ini suatu lompatan akan terjadi bilamana bendera nol dalam keadaan "clear", dan tidak akan terjadi lompatan bilamana bendera nol tersebut dalam keadaan set. Andaikan bahwa JNZ 7800H disimpan pada lokasi 2100H. Jika Z=0, instruksi berikutnya akan berasal dari lokasi 7800H: dan bilamana Z=1, program akan turun ke instruksi urutan berikutnya pada lokasi 2101H.

21. JZ (Jump if Accumulator is Zero)

JZ berfungsi untuk menguji konten-konten akumulator. Jika bukan nol, maka lompatan dilakukan ke alamat relatif yang ditentukan dalam perintah.

22. LCALL (Long Call)

LCALL berfungsi untuk memungkinkan panggilan ke subrutin yang berlokasi dimanapun dalam memori program 64K. Operasi LCALL berjalan seperti berikut:

- Menambahkan ke dalam konter program sebanyak 3, karena perintahnya adalah perintah 3-byte.
- Menambahkan penunjuk stack sebanyak 1.
- Menyimpan byte yang lebih rendah dari konter program ke dalam stack.
- Menambahkan penunjuk stack.
- Menyimpan byte yang lebih tinggi dari program ke dalam stack.
- Me-load konter program dengan alamat tujuan 16-bit.

23. . LJM (Long Jump)

Long Jump berfungsi untuk memungkinkan lompatan tak bersyarat kemana saja dalam lingkup ruang memori program 64K. LCALL adalah perintah 3-byte. Alamat tujuan 16-bit ditentukan secara langsung dalam perintah tersebut. Alamat tujuan ini di-load ke dalam konter program oleh perintah LJM.

24. MOV (Move From Memory)

MOV berfungsi untuk memindahkan isi akumulator/register atau data dari nilai luar atau alamat lain.

25. MOVC (Move From Codec Memory)

Instruksi MOVC berfungsi untuk mengisi accumulator dengan byte kode atau konstanta dari program memory. Alamat byte tersebut adalah hasil penjumlahan unsigned 8 bit pada accumulator dan 16 bit register basis yang dapat berupa data pointer atau program counter. Instruksi ini tidak mempengaruhi flag apapun juga.

26. MOVX (Move Accumulator to External Memory Addressed by Data Pointer)

MOVX berfungsi untuk memindahkan isi akumulator ke memori data eksternal yang alamatnya ditunjukkan oleh isi data pointer.

27. MUL (Multiply)

MUL AB berfungsi untuk mengalikan unsigned 8 bit integer pada accumulator dan register B. Byte rendah (low order) dari hasil perkalian akan disimpan dalam accumulator sedangkan byte tinggi (high order) akan disimpan dalam register B. Jika hasil perkalian lebih besar dari 255 (0FFh), overflow flag akan bernilai '1'. Jika hasil perkalian lebih kecil atau sama dengan 255, overflow flag akan bernilai '0'. Carry flag akan selalu

dikosongkan.

28. NOP (No Operation)

Fungsi NOP adalah eksekusi program akan dilanjutkan ke instruksi berikutnya. Selain PC, instruksi ini tidak mempengaruhi register atau flag apapun juga.

29. ORL (Logical OR Immediate Data to Accumulator)

Instruksi ORL berfungsi sebagai instruksi Gerbang logika OR yang akan menjumlahkan Accumulator terhadap nilai yang ditentukan.

Format : ORL A,#data.

30. POP (Pop Stack to Memory)

Instruksi POP berfungsi untuk menempatkan byte yang ditunjukkan oleh stack pointer ke suatu alamat data.

31. PUSH (Push Memory onto Stack)

Instruksi PUSH berfungsi untuk menaikkan stack pointer kemudian menyimpan isinya ke suatu alamat data pada lokasi yang ditunjuk oleh stack pointer.

32. RET (Return from subroutine)

Intruksi RET berfungsi untuk kembali dari suatu subrutin program ke alamat terakhir subrutin tersebut di panggil.

33. RETI (Return From Interrupt)

RETI berfungsi untuk mengambil nilai byte tinggi dan rendah dari PC dari stack dan mengembalikan kondisi logika interrupt agar dapat menerima interrupt lain dengan prioritas yang sama dengan prioritas interrupt yang baru saja diproses. Stack pointer akan dikurangi dengan 2. Instruksi ini tidak mempengaruhi flag apapun juga. Nilai PSW tidak akan dikembalikan secara otomatis ke kondisi sebelum interrupt. Eksekusi program akan dilanjutkan pada alamat yang diambil tersebut. Umumnya alamat tersebut adalah alamat setelah lokasi dimana terjadi interrupt. Jika interrupt dengan prioritas

sama atau lebih rendah tertunda saat RETI dieksekusi, maka satu instruksi lagi akan dieksekusi sebelum interrupt yang tertunda tersebut diproses.

34. RL (Rotate Accumulator Left)

Instruksi RL berfungsi untuk memutar setiap bit dalam akumulator satu posisi ke kiri.

35. . RLC (Rotate Left through Carry)

Fungsi : Memutar (Rotate) Accumulator ke Kiri (Left) Melalui Carry Flag. Kedelapan bit accumulator dan carry flag akan diputar satu bit ke kiri secara bersama-sama. Bit 7 akan dirotasi ke carry flag, nilai carry flag akan berpindah ke posisi bit 0. Instruksi ini tidak mempengaruhi flag lain.

36. RR (Rotate Right)

Fungsi : Memutar (Rotate) Accumulator ke Kanan (Right). Kedelapan bit accumulator akan diputar satu bit ke kanan. Bit 0 akan dirotasi ke posisi bit 7. Instruksi ini tidak mempengaruhi flag apapun juga.

37. RRC (Rotate Right through Carry)

Fungsi : Memutar (Rotate) Accumulator ke Kanan (Right) Melalui Carry Flag. Kedelapan bit accumulator dan carry flag akan diputar satu bit ke kanan secara bersama-sama. Bit 0 akan dirotasi ke carry flag, nilai carry flag akan berpindah ke posisi bit 7. Instruksi ini tidak mempengaruhi flag lain.

38. SETB (set Carry flag)

Instruksi SETB berfungsi untuk menset carry flag.

39. SJMP (Short Jump)

Sebuah Short Jump berfungsi untuk mentransfer kendali ke alamat tujuan dalam 127 bytes yang mengikuti dan 128 yang mengawali perintah SJMP. Alamat tujuannya ditentukan sebagai sebuah alamat relative 8-bit. Ini adalah Jump tidak bersyarat. Perintah SJMP menambahkan konter program sebanyak 2 dan menambahkan alamat relatif ke dalamnya untuk mendapatkan alamat tujuan. Alamat relatif tersebut ditentukan dalam perintah sebagai 'SJMP rel'.

40. SUBB (Subtract With Borrow)

Fungsi : Pengurangan (Subtract) dengan Peminjaman (Borrow). SUBB mengurangi variabel yang tertera pada operand kedua dan carry flag sekaligus dari accumulator dan menyimpan hasilnya pada accumulator. SUBB akan memberi nilai '1' pada carry flag jika peminjaman ke bit 7 dibutuhkan dan mengosongkan C jika tidak dibutuhkan peminjaman. Jika C bernilai '1' sebelum mengeksekusi SUBB, hal ini menandakan bahwa terjadi peminjaman pada proses pengurangan sebelumnya, sehingga carry flag dan source byte akan dikurangkan dari accumulator secara bersama-sama. AC akan bernilai '1' jika peminjaman ke bit 3 dibutuhkan dan mengosongkan AC jika tidak dibutuhkan peminjaman. OV akan bernilai '1' jika ada peminjaman ke bit 6 namun tidak ke bit 7 atau ada peminjaman ke bit 7 namun tidak ke bit 6. Saat mengurangi signed integer, OV menandakan adanya angka negative sebagai hasil dari pengurangan angka negatif dari angka positif atau adanya angka positif sebagai hasil dari pengurangan angka positif dari

angka negative. Addressing mode yang dapat digunakan adalah: register, direct, register

indirect, atau immediate data.

41. SWAP (Swap Nibbles)

Fungsi : Menukar (Swap) Upper Nibble dan Lower Nibble dalam Accumulator. SWAP A akan menukar nibble (4 bit) tinggi dan nibble rendah dalam accumulator. Operasi ini dapat dianggap sebagai rotasi 4 bit dengan RR atau RL. Instruksi ini tidak mempengaruhi flag apapun juga.

42. XCH (Exchange Bytes)

Fungsi : Menukar (Exchange) Accumulator dengan Variabel Byte. XCH akan mengisi accumulator dengan variabel yang tertera pada operand kedua dan pada saat yang sama juga akan mengisi nilai accumulator ke dalam variabel tersebut. Addressing mode yang dapat digunakan adalah: register, direct, atau register indirect.

43. XCHD (Exchange Digits)

Fungsi : Menukar (Exchange) Digit. XCHD menukar nibble rendah dari accumulator, yang umumnya mewakili angka heksadesimal atau BCD, dengan nibble rendah dari internal data memory yang diakses secara indirect. Nibble tinggi kedua register tidak akan terpengaruh. Instruksi ini tidak mempengaruhi flag apapun juga.

44. XRL (Exclusive OR Logic)

Fungsi : Logika Exclusive OR untuk Variabel Byte XRL akan melakukan operasi bitwise logika exclusive OR antara kedua variabel yang dinyatakan. Hasilnya akan disimpan pada destination byte. Instruksi ini tidak mempengaruhi flag apapun juga. Kedua operand mampu menggunakan enam kombinasi addressing mode. Saat destination byte adalah accumulator, source byte dapat berupa register, direct, register indirect, atau immediate data. Saat destination byte berupa direct address, source byte dapat berupa accumulator atau immediate data.