

SIEMENS



SAB 80C5XX Microcontroller Family

SAB 80C515 / 80C535

SAB 80C517 / 80C537

SAB 80C515A / 83C515A

SAB 80C517A / 83C517A

Pocket Guide 08.98

Instruction List for the SAB 80C51X Microcontroller Family

Kind of Instruction	Mnemonic	Register	Encoding		Affected Flags	Bytes	Cycles	Function	Funktionsbeschreibung	
			Binary	Hex						
Transfer Instructions	Internal RAM	MOV A,Rr	r=0-7	11101 r r r	E8+r	P	1	1	Move register Rr contents to accumulator	Lade Akkumulator mit Inhalt von Register Rr
		MOV Rr,A	r=0-7	11111 r r r	F8+r	-	1	1	Move accumulator contents to register Rr	Lade Register Rr mit Inhalt des Akkumulators
		MOV A,@Ri	i=0-1	1110011 i	E6+i	P	1	1	Move contents of data memory addressed by Ri to accumulator	Lade Akkumulator mit dem Inhalt der internen Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist
		MOV @Ri,A	i=0-1	1111011 i	F6+i	-	1	1	Move accumulator to data memory addressed by Ri	Lade interne Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist, mit dem Inhalt des Akkumulators
		MOV A,dadr *)	-	11100101	E5	P	2	1	Move contents of direct address to accumulator (MOV A,ACC is not a valid instruction!)	Lade Akkumulator mit Inhalt von dadr (MOV A,ACC ist kein gültiger Befehl!)
		MOV dadr,A	-	11110101	F5	-	2	1	Move accumulator contents to direct address	Lade dadr mit Inhalt des Akkumulators
		MOV A,#const8	-	01110100	74	P	2	1	Move immediate data to accumulator	Lade Akkumulator mit 8 Bit Konstante
		MOV Rr,#const8	r=0-7	01111 r r r	78+r	-	2	1	Move immediate data to register	Lade Register Rr mit 8 Bit Konstante
		MOV @Ri,#const8	i=0-1	0111011 i	76+i	-	2	1	Move immediate data to data memory addressed by Ri	Lade interne Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist, mit 8 Bit Konstante
		MOV dadr,#const8	-	01110101	75	-	3	2	Move immediate data to data memory	Lade dadr mit 8 Bit Konstante
		MOV Rr,dadr	r=0-7	10101 r r r	A8+r	-	2	2	Load register Rr with contents of direct address	Lade Register Rr mit Inhalt von dadr
		MOV dadr,Rr	r=0-7	10001 r r r	88+r	-	2	2	Load direct address with contents of register	Lade dadr mit Inhalt von Register Rr
		MOV dadr,@Ri	i=0-1	1000011 i	86+i	-	2	2	Move contents of data memory addressed by Ri to direct address	Lade dadr mit Inhalt der Datenspeicherzelle, die durch Ri adressiert ist
		MOV @Ri,dadr	i=0-1	1010011 i	A6+i	-	2	2	Move data memory to contents of data memory addressed by Ri	Lade interne Datenspeicherzelle, die durch Ri adressiert ist, mit Inhalt von dadr
	MOV dadr1,dadr2	-	10000101	85	-	3	2	Move data memory to data memory	Lade dadr1 mit Inhalt von dadr2	
	MOV DPTR,#const16	-	10010000	90	-	3	2	Move immediate data to data pointer	Lade Daten-Pointer DPTR mit 16 Bit Konstante	
	External RAM	MOVX A,@Ri	i=0-1	1110001 i	E2+i	P	1	2	Move contents of external data memory addressed by Ri to accumulator	Lade Akkumulator mit dem Inhalt der externen Datenspeicherzelle, die durch Ri adressiert ist
		MOVX @Ri,A	i=0-1	1111001 i	F2+i	-	1	2	Move accumulator contents to external data memory, addressed by Ri	Lade externe Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist, mit dem Inhalt des Akkumulators
		MOVX A,@DPTR	-	11100000	E0	P	1	2	Move contents of external data memory addressed by DPTR to accumulator	Lade Akkumulator mit dem Inhalt der externen Datenspeicherzelle, die durch DPTR adressiert ist
		MOVX @DPTR,A	-	11110000	F0	-	1	2	Move accumulator contents to external data memory, addressed by DPTR	Lade externe Datenspeicherzelle, die durch den Inhalt von DPTR adressiert ist, mit dem Inhalt des Akkumulators
	Internal RAM	XCH A,Rr	r=0-7	11001 r r r	C8+r	P	1	1	Exchange accumulator and register contents	Vertausche die Inhalte von Akkumulator und Register Rr
		XCH A,dadr	-	11000101	C5	P	2	1	Exchange accumulator and data memory contents	Vertausche die Inhalte von Akkumulator und dadr
		XCH A,@Ri	i=0-1	1100011 i	C6+i	P	1	1	Exchange accumulator and data memory addressed by Ri	Vertausche die Inhalte von Akkumulator und der internen Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist
		XCHD A,@Ri	i=0-1	1101011 i	D6+i	P	1	1	Exchange low order nibble of accumulator and data memory	Vertausche die Inhalte der niederwertigen Halbbytes von Akkumulator und der internen Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist
		SWAP A	-	11001010	CA	-	1	1	Exchange nibbles of accumulator	Vertausche die Bytehälften des Akkumulators
		PUSH dadr	-	11000000	C0	-	2	2	Increment stack pointer and push data memory onto stack	Der Stack-Pointer wird um 1 erhöht und der Inhalt von dadr wird im Stack abgelegt
POP dadr		-	11010000	D0	-	2	2	Pop data memory from stack and decrement stack pointer	Übertrage den Inhalt der durch den Stack-Pointer adressierten Datenspeicherzelle nach dadr und der Stack-Pointer wird um 1 erniedrigt	
ROM	MOVC A,@A+DPTR	-	10010011	93	P	1	2	Move program memory addressed by accumulator plus DPTR to accumulator	Lade Akkumulator mit dem Inhalt der Programmspeicherzelle, die durch die Summe von DPTR und Akkumulator adressiert ist	
	MOVC A,@A+PC	-	10000011	83	P	1	2	Move program memory addressed by accumulator plus program counter to accumulator	Lade Akkumulator mit dem Inhalt der Programmspeicherzelle, die durch die Summe von Programmzähler und Akkumulator adressiert ist	

Kind of Instruction	Mnemonic		Register	Encoding		Affected Flags	Bytes	Cycles	Function	Funktionsbeschreibung
				Binary	Hex					
Boolean Instructions	AND	ANL A,Rr	r=0-7	01011 r r r	58+r	P	1	1	Logical AND accumulator with register contents to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der UND-Verknüpfung von Akkumulator und dem Inhalt von Register Rr
		ANL A,@Ri	i=0-1	0101011 i	56+i	P	1	1	Logical AND accumulator with data memory content addressed by Ri to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der UND-Verknüpfung von Akkumulator und dem Inhalt der internen Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist
		ANL A,dadr	-	01010101	55	P	2	1	Logical AND accumulator with contents of direct address to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der UND-Verknüpfung von Akkumulator und dem Inhalt von dadr
		ANL A,#const8	-	01010100	54	P	2	1	Logical AND accumulator with immediate data to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der UND-Verknüpfung von Akkumulator und der 8 Bit Konstanten
		ANL dadr,A	-	01010010	52	-	2	1	Logical AND accumulator with contents of direct address to direct address	Der Inhalt von dadr wird überschrieben durch das Ergebnis der UND-Verknüpfung von Akkumulator und dem Inhalt von dadr
		ANL dadr,#const8	-	01010011	53	-	3	2	Logical AND contents of direct address and immediate data to direct address	Der Inhalt von dadr wird überschrieben durch das Ergebnis der UND-Verknüpfung der 8 Bit Konstanten mit dem Inhalt von dadr
	OR	ORL A,Rr	r=0-7	01001 r r r	48+r	P	1	1	Logical OR accumulator with register contents to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Akkumulator und dem Inhalt von Register Rr
		ORL A,@Ri	i=0-1	0100011 i	46+i	P	1	1	Logical OR accumulator with data memory contents addressed by Ri to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Akkumulator und dem Inhalt der internen Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist
		ORL A,dadr	-	01000101	45	P	2	1	Logical OR accumulator with contents of direct address to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Akkumulator und dem Inhalt von dadr
		ORL A,#const8	-	01000100	44	P	2	1	Logical OR accumulator with immediate data to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Akkumulator und der 8 Bit Konstanten
		ORL dadr,A	-	01000010	42	-	2	1	Logical OR accumulator with contents of direct address to direct address	Der Inhalt von dadr wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Akkumulator und dem Inhalt von dadr
		ORL dadr,#const8	-	01000011	43	-	3	2	Logical OR contents of direct address and immediate data to direct address	Der Inhalt von dadr wird überschrieben durch das Ergebnis der ODER-Verknüpfung der 8 Bit Konstanten und dem Inhalt von dadr
	EXCLUSIVE OR	XRL A,Rr	r=0-7	01101 r r r	68+r	P	1	1	Logical XOR accumulator with register contents to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der EXODER-Verknüpfung von Akkumulator und dem Inhalt von Register Rr
		XRL A,@Ri	i=0-1	0110011 i	66+i	P	1	1	Logical XOR accumulator with data memory contents addressed by Ri to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der EXODER-Verknüpfung von Akkumulator und dem Inhalt der internen Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist
		XRL A,dadr	-	01100101	65	P	2	1	Logical XOR accumulator with contents of direct address to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der EXODER-Verknüpfung von Akkumulator und dem Inhalt von dadr
		XRL A,#const8	-	01100100	64	P	2	1	Logical XOR accumulator with immediate mask	Der Akkumulator wird überschrieben durch das Ergebnis der EXODER-Verknüpfung von Akkumulator und der 8 Bit Konstanten
		XRL dadr,A	-	01100010	62	-	2	1	Logical XOR accumulator with contents of direct address to direct address	Der Inhalt von dadr wird überschrieben durch das Ergebnis der EXODER-Verknüpfung von Akkumulator und dem Inhalt von dadr
		XRL dadr,#const8	-	01100011	63	-	3	2	Logical XOR contents of direct address and immediate data to direct address	Der Inhalt von dadr wird überschrieben durch das Ergebnis der EXODER-Verknüpfung der 8 Bit Konstanten und dem Inhalt von dadr
Instructions for Bit Manipulations	Transfer	MOV C,badr	-	10100010	A2	CY	2	1	Move contents of bit address to carry	Lade Carry mit Inhalt von badr
		MOV badr,C	-	10010010	92	-	2	2	Move content of carry to bit address	Lade badr mit Inhalt des Carry
	AND	ANL C,badr	-	10000010	82	CY	2	2	Logical AND carry with contents of bit address	Das Carry wird überschrieben durch das Ergebnis der UND-Verknüpfung von Carry und Inhalt von badr
		ANL C,/badr	-	10110000	B0	CY	2	2	Logical AND carry with complement of contents of bit address	Das Carry wird überschrieben durch das Ergebnis der UND-Verknüpfung von Carry und dem invertierten Inhalt von badr
	OR	ORL C,badr	-	01110010	72	CY	2	2	Logical OR carry with contents of bit address	Das Carry wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Carry und Inhalt von badr
		ORL C,/badr	-	10100000	A0	CY	2	2	Logical OR carry with complement of contents of bit address	Das Carry wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Carry und dem invertierten Inhalt von badr
	Clear	CLR C	-	11000011	C3	CY	1	1	Clear carry	Carry löschen
		CLR badr	-	11000010	C2	-	2	1	Clear contents of bit address	Lösche Inhalt von badr
	Complement	CPL C	-	10110011	B3	CY	1	1	Complement carry	Carry invertieren
		CPL badr	-	10110010	B2	-	2	1	Complement contents of bit address	Invertiere Inhalt von badr
Set	SETB C	-	11010011	D3	CY	1	1	Set carry	Carry setzen	
	SETB badr	-	11010010	D2	-	2	1	Set contents of bit address	Setze Inhalt von badr	

Kind of Instruction	Mnemonic	Register	Encoding		Affected Flags	Bytes	Cycles	Function	Funktionsbeschreibung	
			Binary	Hex						
Arithmetic Instructions	ADD	ADD A,Rr	r=0-7	00101 r r r	28+r	CY,AC,OV,P	1	1	Add register contents to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akkumulator und Inhalt des Registers Rr
		ADD A,@Ri	i=0-1	0010011 i	26+i	CY,AC,OV,P	1	1	Add contents of data memory addressed by Ri to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akkumulator und Inhalt der internen Datenspeicherzelle, die durch Ri adressiert ist
		ADD A,dadr	-	00100101	25	CY,AC,OV,P	2	1	Add contents of direct address to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akkumulator und Inhalt von dadr
		ADD A,#const8	-	00100100	24	CY,AC,OV,P	2	1	Add immediate data to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akkumulator und der 8 Bit Konstanten
		ADDC A,Rr	r=0-7	00111 r r r	38+r	CY,AC,OV,P	1	1	Add carry and register contents to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akkumulator, dem Inhalt des Registers Rr und dem Inhalt des Carry-Flags
		ADDC A,@Ri	i=0-1	0011011 i	36+i	CY,AC,OV,P	1	1	Add carry and contents of data memory addressed by Ri to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akkumulator, dem Inhalt der internen Datenspeicherzelle, die durch Ri adressiert ist, und dem Inhalt des Carry-Flags
		ADDC A,dadr	-	00110101	35	CY,AC,OV,P	2	1	Add carry and contents of direct address to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akkumulator, dem Inhalt von dadr und dem Inhalt des Carry-Flags
		ADDC A,#const8	-	00110100	34	CY,AC,OV,P	2	1	Add carry and immediate data to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akkumulator, der 8 Bit Konstanten und dem Inhalt des Carry-Flags
		INC A	-	00000100	04	P	1	1	Increment accumulator	Zum Inhalt des Akkumulators wird 1 addiert
		INC Rr	r=0-7	00001 r r r	08+r	-	1	1	Increment register	Zum Inhalt des Registers Rr wird 1 addiert
		INC @Ri	i=0-1	0000011 i	06+i	-	1	1	Increment data memory contents addressed by Ri	Zum Inhalt der internen Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist, wird 1 addiert
		INC dadr	-	00000101	05	-	2	1	Increment contents of direct address	Zum Inhalt von dadr wird 1 addiert
	INC DPTR	-	10100011	A3	-	1	2	Increment data pointer	Zum Inhalt des Registers DPTR wird 1 addiert	
	DA A	-	11010100	D4	CY,P	1	1	Decimal adjust accumulator for addition	Das Ergebnis einer vorangegangenen Addition zweier BCD-Zahlen wird unter Berücksichtigung von Carry und Hilfcarry wieder zu 2 BCD-Ziffern korrigiert	
Subtract	SUBB A,Rr	r=0-7	10011 r r r	98+r	CY,AC,OV,P	1	1	Subtract sum of register and carry from accumulator	Die Summe der Inhalte von Carry und Register Rr wird vom Akkumulator subtrahiert. Das Ergebnis überschreibt den Akkumulator.	
	SUBB A,@Ri	i=0-1	1001011 i	96+i	CY,AC,OV,P	1	1	Subtract sum of contents of data memory addressed by Ri and carry from accumulator	Die Summe der Inhalte von Carry und der internen Datenspeicherzelle, die durch Ri adressiert ist, wird vom Akkumulator subtrahiert. Das Ergebnis überschreibt den Akkumulator.	
	SUBB A,dadr	-	10010101	95	CY,AC,OV,P	2	1	Subtract sum of contents of direct address and carry from accumulator	Die Summe der Inhalte von Carry und der Adresse dadr wird vom Akkumulator subtrahiert. Das Ergebnis überschreibt den Akkumulator.	
	SUBB A,#const8	-	10010100	94	CY,AC,OV,P	2	1	Subtract sum of immediate data and carry from accumulator	Die Summe der Inhalte von Carry und der 8 Bit Konstanten wird vom Akkumulator subtrahiert. Das Ergebnis überschreibt den Akkumulator.	
	DEC A	-	00010100	14	P	1	1	Decrement accumulator	Vom Inhalt des Akkumulators wird 1 subtrahiert	
	DEC Rr	r=0-7	00011 r r r	18+r	-	1	1	Decrement register	Vom Inhalt des Registers Rr wird 1 subtrahiert	
	DEC @Ri	i=0-1	0001011 i	16+i	-	1	1	Decrement contents of data memory addressed by Ri	Vom Inhalt der internen Datenspeicherzelle, die durch Ri adressiert ist, wird 1 subtrahiert	
	DEC dadr	-	00010101	15	-	2	1	Decrement contents of direct address	Vom Inhalt der Adresse dadr wird 1 subtrahiert	
Multiply	MUL AB	-	10100100	A4	CY,OV,P	1	4	Multiply accumulator with B-register	Die Inhalte von Akkumulator und B-Register werden multipliziert. Das niederwertige Byte des Produktes überschreibt den Akkumulator, das höherwertige Byte das B-Register. Das Carry wird gelöscht, das OV-Bit wird gesetzt, wenn das Ergebnis im B-Register 0 ist.	
Divide	DIV AB	-	10000100	84	CY,OV,P	1	4	Divide accumulator by B-register	Der Inhalt des Akkumulators wird durch den Inhalt des B-Registers dividiert. Der Quotient überschreibt den Akkumulator, der Rest das B-Register. Das Carry wird gelöscht. Bei Division durch 0 wird das OV-Bit gesetzt.	
Shift Instructions	RL A	-	00100011	23	-	1	1	Rotate left without carry	Verschiebe den Inhalt des Akkumulators um 1 Stelle nach links. Die Stelle 2^7 wird zur Stelle 2^0 verschoben.	
	RLC A	-	00110011	33	CY,P	1	1	Rotate left through carry	Verschiebe den Inhalt des Akkumulators um 1 Stelle nach links über das Carry. Das Carry wird an die Stelle 2^0 verschoben.	
	RR A	-	00000011	03	-	1	1	Rotate right without carry	Verschiebe den Inhalt des Akkumulators um 1 Stelle nach rechts. Die Stelle 2^0 wird zur Stelle 2^7 verschoben.	
	RRC A	-	00010011	13	CY,P	1	1	Rotate right through carry	Verschiebe den Inhalt des Akkumulators um 1 Stelle nach rechts über das Carry. Das Carry wird an die Stelle 2^7 verschoben.	

Kind of Instruction	Mnemonic	Register	Encoding		Affected Flags	Bytes	Cycles	Function	Funktionsbeschreibung	
			Binary	Hex						
Branch Instructions	Unconditional Jumps	LJMP adr16	-	00000010	02	-	3	2	Jump to absolute address	Setze das Programm bei der 16 Bit Adresse adr16 fort
		SJMP rel	-	10000000	80	-	2	2	Jump to relative address	Setze das Programm relativ zum Programmzähler fort
		AJMP adr11	-	See Table 'Branch instructions'		-	2	2	Jump to absolute address	Setze das Programm bei der 11 Bit Adresse adr11 innerhalb der 2K, auf die der Programmzähler zeigt, fort
		JMP @A+DPTR	-	01110011	73	-	1	2	Jump to sum of accumulator and data pointer	Setze das Programm an der Adresse fort, die sich aus der Summe von Akkumulator und DPTR ergibt
	Conditional Jumps	JC rel	-	01000000	40	-	2	2	Jump if carry is set	Springe relativ zum Programmzähler, wenn Carry = 1 ist
		JNC rel	-	01010000	50	-	2	2	Jump if carry is not set	Springe relativ zum Programmzähler, wenn Carry = 0 ist
		JB badr,rel	-	00100000	20	-	3	2	Jump if bit is set	Springe relativ zum Programmzähler, wenn der Inhalt von badr=1 ist
		JNB badr,rel	-	00110000	30	-	3	2	Jump if bit is not set	Springe relativ zum Programmzähler, wenn der Inhalt von badr=0 ist
		JBC badr,rel	-	00010000	10	-	3	2	Jump and clear if bit is set	Springe relativ zum Programmzähler, wenn der Inhalt von badr=1 ist und lösche den Inhalt von badr
		JZ rel	-	01100000	60	-	2	2	Jump relative if accumulator is zero	Springe relativ zum Programmzähler, wenn der Akkumulator = 0 ist
		JNZ rel	-	01110000	70	-	2	2	Jump relative if accumulator is not zero	Springe relativ zum Programmzähler, wenn der Akkumulator ≠ 0 ist
		CJNE A,dadr,rel	-	10110101	B5	CY	3	2	Compare memory to accumulator. Jump if not equal	Springe relativ zum Programmzähler, wenn die Inhalte von Akkumulator und dadr ungleich sind. Das Carry wird gesetzt, wenn dadr größer ist, sonst wird es rückgesetzt.
		CJNE A,#const8,rel	-	10110100	B4	CY	3	2	Compare immediate data to accumulator. Jump if not equal	Springe relativ zum Programmzähler, wenn der Inhalt des Akkumulators ungleich const8 ist. Das Carry wird gesetzt, wenn const8 größer ist, sonst wird es rückgesetzt.
		CJNE Rr,#const8,rel	r=0-7	10111 r r r	B8+r	CY	3	2	Compare immediate data to register. Jump if not equal	Springe relativ zum Programmzähler, wenn der Inhalt von Rr ungleich const8 ist. Das Carry wird gesetzt, wenn const8 größer ist, sonst wird es rückgesetzt.
	CJNE @Ri,#const8,rel	i=0-1	1011011 i	B6+i	CY	3	2	Compare constants of indirect address to immediate data. Jump if not equal	Springe relativ zum Programmzähler, wenn der Inhalt der Datenspeicherzelle, die durch Ri adressiert ist, ungleich const8 ist. Das Carry wird gesetzt, wenn const8 größer ist, sonst wird es rückgesetzt.	
	DJNZ Rr,rel	r=0-7	11011 r r r	D8+r	-	2	2	Decrement register and jump if not zero	Der Inhalt von Register Rr wird um 1 erniedrigt. Ist dann der Inhalt ≠ 0, springe relativ zum Programmzähler	
	DJNZ dadr,rel	-	11010101	D5	-	3	2	Decrement memory and jump if not zero	Der Inhalt von dadr wird um 1 erniedrigt. Ist dann der Inhalt ≠ 0, springe relativ zum Programmzähler	
	Unconditional Calls & Returns	LCALL adr16	-	00010010	12	-	3	2	Long call to adr16	Der Stack-Pointer wird erhöht. Der Inhalt des Programmzählers wird im Stack abgelegt. Das Programm wird bei der Adresse adr16 fortgesetzt.
		ACALL adr11	-	See Table 'Branch instructions'		-	2	2	Absolute call within current 2 K	Der Stack-Pointer wird erhöht. Der Inhalt des Programmzählers wird im Stack abgelegt. Das Programm wird bei der 11 Bit Adresse adr11 innerhalb der 2 K fortgesetzt, auf die der Programmzähler zeigt.
RET		-	00100010	22	-	1	2	Return from subroutine	Die oberen zwei Byte des Stack werden in den Programmzähler geladen. Der Stack-Pointer wird dabei erniedrigt.	
RETI		-	00110010	32	-	1	2	Return from interrupt routine	Die oberen zwei Byte des Stack werden in den Programmzähler geladen. Der Stack-Pointer wird dabei erniedrigt. Die aktuelle Interruptebene wird freigegeben	
	NOP	-	00000000	00	-	1	1	No operation	Leerbefehl	

Table "Branch Instructions"

Page *)	AJMP		ACALL	
	aaa 00001		aaa 10001	
	Binary	Hex	Binary	Hex
0	000 00001	01	000 10001	11
1	001 00001	21	001 10001	31
2	010 00001	41	010 10001	51
3	011 00001	61	011 10001	71
4	100 00001	81	100 10001	91
5	101 00001	A1	101 10001	B1
6	110 00001	C1	110 10001	D1
7	111 00001	E1	111 10001	F1

*) One page represents the range 00 ... FF addressed by the low byte of a 16-bit address.

Interrupt Vectors

Address	Interrupt Source	80C515	80C515A	80C517	80C517A
0000H	Start address after Reset	X	X	X	X
0003H	External interrupt 0	X	X	X	X
000BH	Timer 0 overflow interrupt	X	X	X	X
0013H	External interrupt 1	X	X	X	X
001BH	Timer 1 overflow interrupt	X	X	X	X
0023H	Serial channel interrupt Serial channel 0 interrupt	X	X	X	X
002BH	Timer 2 overflow / external reload interrupt	X	X	X	X
0043H	A/D converter interrupt	X	X	X	X
004BH	External interrupt 2 External interrupt 2 / Compare event with CC4	X	X	X	X
0053H	External interrupt 3 / Compare event with CRC	X	X	X	X
005BH	External interrupt 4 / Compare event with CC1	X	X	X	X
0063H	External interrupt 5 / Compare event with CC2	X	X	X	X
006BH	External interrupt 6 / Compare event with CC3	X	X	X	X
0083H	Serial channel 1 interrupt			X	X
0093H	CMx compare register interrupt				X
009BH	Compare timer overflow interrupt				X
00A3H	Compare set interrupt				X
00ABH	Compare clear interrupt				X

"X" means that an interrupt is available for the corresponding microcontroller.

Instruction List in Hexadecimal Order

Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic
00	NOP	20	JB	40	JC rel	60	JZ rel	80	SJMP rel	A0	ORL C,/badr	C0	PUSH dadr	E0	MOVX A,@DPTR
01	AJMP page 0	21	AJMP page 1	41	AJMP page 2	61	AJMP page 3	81	AJMP page 4	A1	AJMP page 5	C1	AJMP page 6	E1	AJMP page 7
02	LJMP adr1 6	22	RET	42	ORL dadr,A	62	XRL dadr,A	82	ANL	A2	MOV C,badr	C2	CLR C	E2	MOVX A,@R0
03	RR A	23	RL A	43	ORL dadr,#c8	63	XRL dadr,#c8	83	MOVC A,@A+PC	A3	INC DPTR	C3	CLR A	E3	MOVX A,@R1
04	INC A	24	ADD A,#c8	44	ORL A,#c8	64	XRL A,#c8	84	DIV AB	A4	MUL AB	C4	SWAP A	E4	CLR A
05	INC dadr	25	ADD A,dadr	45	ORL A,dadr	65	XRL A,dadr	85	MOV dadr,dadr	A5	-	C5	XCH a,dadr	E5	MOV A,dadr
06	INC @R0	26	ADD A,@R0	46	ORL A,@R0	66	XRL A,@R0	86	MOV dadr,@R0	A6	MOV @R0,dadr	C6	XCH A,@R0	E6	MOV A,@R0
07	INC @R1	27	ADD A,@R1	47	ORL A,@R1	67	XRL A,@R1	87	MOV dadr,@R1	A7	MOV @R1,dadr	C7	XCH A,@R1	E7	MOV A,@R1
08	INC R0	28	ADD A,R0	48	ORL A,R0	68	XRL A,R0	88	MOV dadr,R0	A8	MOV R0,dadr	C8	XCH A,R0	E8	MOV A,R0
09	INC R1	29	ADD A,R1	49	ORL A,R1	69	XRL A,R1	89	MOV dadr,R1	A9	MOV R1,dadr	C9	XCH A,R1	E9	MOV A,R1
0A	INC R2	2A	ADD A,R2	4A	ORL A,R2	6A	XRL A,R2	8A	MOV dadr,R2	AA	MOV R2,dadr	CA	XCH A,R2	EA	MOV A,R2
0B	INC R3	2B	ADD A,R3	4B	ORL A,R3	6B	XRL A,R3	8B	MOV dadr,R3	AB	MOV R3,dadr	CB	XCH A,R3	EB	MOV A,R3
0C	INC R4	2C	ADD A,R4	4C	ORL A,R4	6C	XRL A,R4	8C	MOV dadr,R4	AC	MOV R4,dadr	CC	XCH A,R4	EC	MOV A,R4
0D	INC R5	2D	ADD A,R5	4D	ORL A,R5	6D	XRL A,R5	8D	MOV dadr,R5	AD	MOV R5,dadr	CD	XCH A,R5	ED	MOV A,R5
0E	INC R6	2E	ADD A,R6	4E	ORL A,R6	6E	XRL A,R6	8E	MOV dadr,R6	AE	MOV R6,dadr	CE	XCH A,R6	EE	MOV A,R6
0F	INC R7	2F	ADD A,R7	4F	ORL A,R7	6F	XRL A,R7	8F	MOV dadr,R7	AF	MOV R7,dadr	CF	XCH A,R7	EF	MOV A,R7
10	JBC badr,rel	30	JNB badr,rel	50	JNC rel	70	JNZ rel	90	MOV DPTR	B0	ANL C,/badr	D0	POP dadr	F0	MOVX @DPTR,A
11	ACALL page 0	31	ACALL page 1	51	ACALL page 2	71	ACALL page 3	91	ACALL page 4	B1	ACALL page 5	D1	ACALL page 6	F1	ACALL page 7
12	LCALL adr16	32	RETI	52	ANL dadr,A	72	ORL C;badr	92	MOV badr,C	B2	CPL badr	D2	SETB badr	F2	MOVX @R0,A
13	RRC A	33	RLC A	53	ANL dadr,#c8	73	JMP @A+DPTR	93	MOVC A,@A+DPTR	B3	CPL C	D3	SETB C	F3	MOVX @R1,A
14	IDEC A	34	ADDC A,#c8	54	ANL A,#c8	74	MOV A,#c8	94	SUBB A,#c8	B4	CJNE A,#c8,rel	D4	DA A	F4	CPL A
15	DEC dadr	35	ADDC A,dadr	55	ANL A,dadr	75	MOV dadr,#c8	95	SUBB A,dadr	B5	CJNE A,dadr,rel	D5	DJNZ dadr,rel	F5	MOV dadr,A
16	DEC @R0	36	ADDC A,@R0	56	ANL A,@R0	76	MOV @R0,#c8	96	SUBB A,@R0	B6	CJNE @R0,#c8,rel	D6	XCHD A,@R0	F6	MOV @R0,A
17	DEC @R1	37	ADDC A,@R1	57	ANL A,@R1	77	MOV @R1,#c8	97	SUBB A,@R1	B7	CJNE @R1,#c8,rel	D7	XCHD A,@R1	F7	MOV @R1,A
18	DEC R0	38	ADDC A,R0	58	ANL A,R0	78	MOV R0,#c8	98	SUBB A,R0	B8	CJNE R0,#c8,rel	D8	DJNZ R0,rel	F8	MOV R0,A
19	DEC R1	39	ADDC A,R1	59	ANL A,R1	79	MOV R1,#c8	99	SUBB A,R1	B9	CJNE R1,#c8,rel	D9	DJNZ R1,rel	F9	MOV R1,A
1A	IDEC R2	3A	ADDC A,R2	5A	ANL A,R2	7A	MOV R2,#c8	9A	SUBB A,R2	BA	CJNE R2,#c8,rel	DA	DJNZ R2,rel	FA	MOV R2,A
1B	IDEC R3	3B	ADDC A,R3	5B	ANL A,R3	7B	MOV R3,#c8	9B	SUBB A,R3	BB	CJNE R3,#c8,rel	DB	DJNZ R3,rel	FB	MOV R3,A
1C	IDEC R4	3C	ADDC A,R4	5C	ANL A,R4	7C	MOV R4,#c8	9C	SUBB A,R4	BC	CJNE R4,#c8,rel	DC	DJNZ R4,rel	FC	MOV R4,A
1D	DEC R5	3D	ADDC A,R5	5D	ANL A,R5	7D	MOV R5,#c8	9D	SUBB A,R5	BD	CJNE R5,#c8,rel	DD	DJNZ R5,rel	FD	MOV R5,A
1E	DEC R6	3E	ADDC A,R6	5E	ANL A,R6	7E	MOV R6,#c8	9E	SUBB A,R6	BE	CJNE R6,#c8,rel	DE	DJNZ R6,rel	FE	MOV R6,A
1F	DEC R7	3F	ADDC A,R7	5F	ANL A,R7	7F	MOV R7,#c8	9F	SUBB A,R7	BF	CJNE R7,#c8,rel	DF	DJNZ R7,rel	FF	MOV R7,A

c8 = const8 c16=const16

ASCII-Code Table

Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII
00	NUL	10	DLE	20	SP	30	0	40	@	50	P	60	'	70	p
01	SOH	11	DC1 (X-ON)	21	!	31	1	41	A	51	Q	61	a	71	q
02	STX	12	DC2 (TAPE)	22	"	32	2	42	B	52	R	62	b	72	r
03	ETX	13	DC3 (X-OFF)	23	#	33	3	43	C	53	S	63	c	73	s
04	EOT	14	DC4 (TAPE)	24	\$	34	4	44	D	54	T	64	d	74	t
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e	75	u
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	v
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g	77	w
08	BS	18	CAN	28	(38	8	48	H	58	X	68	h	78	x
09	HT (TAB)	19	EM	29)	39	9	49	I	59	Y	69	i	79	y
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[6B	k	7B	{
0C	FF	1C	FS	2C	,	3C	<	4C	L	5C	\	6C	l	7C	
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D]	D	m	7D	} (ALT MODE)
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL(RUBOUT)

Special Function Registers in Order of Their Addresses

Helevtica narrow (6.5)
 Helevtica narrow (7.0)
 Helevtica (6.5)
 Helevtica (7.0)

		.7 .6 .5 .4 .3 .2 .1 .0				- - - - -				.1 .0													
Hex	Name	80C515	80C515A	80C517	80C517A	Hex	Name	80C515	80C515A	80C517	80C517A	Hex	Name	80C515	80C515A	80C517	80C517A						
80	P0	X	X	X	X	A0	P2	X	X	X	X	C0	IRCON IRCON0	X	X	X	X	E0	ACC	X	X	X	X
81	SP	X	X	X	X	A1	COMSETL				X	C1	CCEN	X	X	X	X	E1	CTCON			X	X
82	DPL	X	X	X	X	A2	COMSETH				X	C2	CCL1	X	X	X	X	E2	CML3			X	X
83	DPH	X	X	X	X	A3	COMCLRL				X	C3	CCH1	X	X	X	X	E3	CMH3			X	X
84	WDTL				X	A4	COMCLRH				X	C4	CCL2	X	X	X	X	E4	CML4			X	X
85	WDTH				X	A5	SETMSK				X	C5	CCH2	X	X	X	X	E5	CMH4			X	X
86	WDTREL		X	X	X	A6	CLRMSK				X	C6	CCL3	X	X	X	X	E6	CML5			X	X
87	PCON	X	X	X	X	A7						C7	CCH3	X	X	X	X	E7	CMH5			X	X
88	TCON	X	X	X	X	A8	IEN0	X	X	X	X	C8	T2CON	X	X	X	X	E8	P4	X	X	X	X
89	TMOD	X	X	X	X	A9	IP0	X	X	X	X	C9	CC4EN			X	X	E9	MD0			X	X
8A	TL0	X	X	X	X	AA	SRELL SORELL		X		X	CA	CRCL	X	X	X	X	EA	MD1			X	X
8B	TL1	X	X	X	X	AB						CB	CRCH	X	X	X	X	EB	MD2			X	X
8C	TH0	X	X	X	X	AC						CC	TL2	X	X	X	X	EC	MD3			X	X
8D	TH1	X	X	X	X	AD						CD	TH2	X	X	X	X	ED	MD4			X	X
8E						AE						CE	CCL4			X	X	EE	MD5			X	X
8F						AF						CF	CCH4			X	X	EF	ARCON			X	X
90	P1	X	X	X	X	B0	P3	X	X	X	X	D0	PSW	X	X	X	X	F0	B	X	X	X	X
91	XPAGE		X		X	B1	SYSCON		X		X	D1	IRCON1				X	F1					
92	DPSEL			X	X	B2						D2	CML0			X	X	F2	CML6			X	X
93						B3						D3	CMH0			X	X	F3	CMH6			X	X
94						B4						D4	CML1			X	X	F4	CML7			X	X
95						B5						D5	CMH1			X	X	F5	CMH7			X	X
96						B6						D6	CML2			X	X	F6	CMEN			X	X
97						B7						D7	CMH2			X	X	F7	CMSEL			X	X
98	SCON S0CON	X	X	X	X	B8	IEN1	X	X	X	X	D8	ADCON ADCON0	X	X	X	X	F8	P5	X	X	X	X
99	SBUF S0BUF	X	X	X	X	B9	IP1	X	X	X	X	D9	ADDAT ADDATH	X	X	X	X	F9					
9A	IEN2			X	X	BA	SRELL S0RELLH		X		X	DA	DAPR ADDATL	X	X	X	X	FA	P6			X	X
9B	S1CON			X	X	BB	S1RELLH			X	X	DB	P6 P7	X	X	X	X	FB					
9C	S1BUF			X	X	BC						DC	ADCON1		X	X	X	FC					
9D	S1RELL			X	X	BD						DD	P8			X	X	FD					
9E						BE						DE	CTRELL			X	X	FE					
9F						BF						DF	CTRELLH			X	X	FF					

Special Function Registers in Functional Groups

CPU

E0 _H	.7	.6	.5	.4	.3	.2	.1	.0	ACC
F0 _H	.7	.6	.5	.4	.3	.2	.1	.0	B
D0 _H	CY	AC	F0	RS1	RS0	OV	F1	P	PSW
81 _H	.7	.6	.5	.4	.3	.2	.1	.0	SP
82 _H	.7	.6	.5	.4	.3	.2	.1	.0	DPL
83 _H	.7	.6	.5	.4	.3	.2	.1	.0	DPH
92 _H	-	-	-	-	-	.2	.1	.0	DPSEL 2) 3) 4)
91 _H	.7	.6	.5	.4	.3	.2	.1	.0	XPAGE 2) 4)
B1 _H	-	-	-	-	-	-	XMAP	XMAP	SYSCON 2) 4)
							1	0	
D8 _H	BD	CLK	-	BSY	ADM	MX2	MX1	MX0	ADCON 1)
D8 _H	BD	CLK	ADEX	BSY	ADM	MX2	MX1	MX0	ADCON0 2) 3) 4)

Timer 0/1

88 _H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON
89 _H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	TMOD
8A _H	.7	.6	.5	.4	.3	.2	.1	.0	TL0
8B _H	.7	.6	.5	.4	.3	.2	.1	.0	TL1
8C _H	.7	.6	.5	.4	.3	.2	.1	.0	TH0
8D _H	.7	.6	.5	.4	.3	.2	.1	.0	TH1

Interrupt System

A8 _H	EAL	WDT	ET2	ES	ET1	EX1	ET0	EX0	IE0 1) 2)
A8 _H	EAL	WDT	ET2	ES0	ET1	EX1	ET0	EX0	IE0 3) 4)
B8 _H	EX EN2	SWDT	EX6	EX5	EX4	EX3	EX2	EADC	IE1
9A _H	-	-	-	-	ECT	-	-	ES1	IE2 3)
9A _H	-	-	ECR	ECS	ECT	ECMP	-	ES1	IE2 4)
A9 _H	-	WDTS	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	IP0 1)
A9 _H	OWDS	WDTS	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	IP0 2) 3) 4)
B9 _H	-	-	IP1.5	IP1.4	IP1.3	IP1.2	IP1.1	IP1.0	IP1
88 _H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON
C8 _H	T2PS	I3FR	I2FR	T2R1	T2R0	T2CM	T2I1	T2I0	T2CON
C0 _H	EXF2	TF2	IEX6	IEX5	IEX4	IEX3	IEX2	IADC	IRCON 1-3) IRCON0 4)
D1 _H	ICMP	ICMP	ICMP	ICMP	ICMP	ICMP	ICMP	ICMP	IRCON1 4)
	7	6	5	4	3	2	1	0	
E1 _H	T2PS1	-	-	-	CTF	CLK2	CLK1	CLK0	CTCON 3)

Serial Channels

98 _H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	SCON 1) 2)
98 _H	SM0	SM1	SM2	REN0	TB80	RB80	TI0	RI0	SOCON 3) 4)
99 _H	.7	.6	.5	.4	.3	.2	.1	.0	SBUF 1)2) SOBUF 3)4)
9B _H	SM	-	SM2	REN1	TB81	RB81	TI1	RI1	S1CON 3) 4)
9C _H	.7	.6	.5	.4	.3	.2	.1	.0	S1BUF 3) 4)
AA _H	.7	.6	.5	.4	.3	.2	.1	.0	SRELL 2) SORELL 3) 4)
BA _H	-	-	-	-	-	-	.1	.0	SRELLH 2) SORELLH 3) 4)
9D _H	.7	.6	.5	.4	.3	.2	.1	.0	S1RELL 3) 4)
BB _H	-	-	-	-	-	-	.1	.0	S1RELLH 3) 4)
D8 _H	BD	CLK	-	BSY	ADM	MX2	MX1	MX0	ADCON 1)
D8 _H	BD	CLK	ADEX	BSY	ADM	MX2	MX1	MX0	ADCON0 2) 3) 4)
87 _H	SMOD	PDS	IDLS	SD	GF1	GF0	PDE	IDLE	PCON

A/D Converter

D8 _H	BD	CLK	-	BSY	ADM	MX2	MX1	MX0	ADCON 1)
D8 _H	BD	CLK	ADEX	BSY	ADM	MX2	MX1	MX0	ADCON0 2) 3) 4)
DC _H	-	-	-	-	MX3	MX2	MX1	MX0	ADCON1 1)
DC _H	ADCL	-	-	-	MX3	MX2	MX1	MX0	ADCON1 2) 4)
D9 _H	.7	.6	.5	.4	.3	.2	.1	.0	ADDAT 1)3) ADDATH 2) 4)
DA _H	.7	.6	-	-	-	-	-	-	ADDATL 2) 4)
DA _H	.7	.6	.5	.4	.3	.2	.1	.0	DAPR 1) 3)

MUL/DIV Unit 3) and 4) only

EF _H	MDEF	MDOV	SLR	SC.4	SC.3	SC.2	SC.1	SC.0	ARCON
E9 _H	.7	.6	.5	.4	.3	.2	.1	.0	MD0
EA _H	.7	.6	.5	.4	.3	.2	.1	.0	MD1
EB _H	.7	.6	.5	.4	.3	.2	.1	.0	MD2
EC _H	.7	.6	.5	.4	.3	.2	.1	.0	MD3
ED _H	.7	.6	.5	.4	.3	.2	.1	.0	MD4
EE _H	.7	.6	.5	.4	.3	.2	.1	.0	MD5

Ports

80 _H	.7	.6	.5	.4	.3	.2	.1	.0	P0
90 _H	T2	CLK OUT	T2EX	INT2	INT6	INT5	INT4	INT3	P1
A0 _H	.7	.6	.5	.4	.3	.2	.1	.0	P2

Timer 2 / Comapre/Capture Unit

C8 _H	T2PS	I3FR	I2FR	T2R1	T2R0	T2CM	T2I1	T2I0	T2CON
CC _H	.7	.6	.5	.4	.3	.2	.1	.0	TL2
CD _H	.7	.6	.5	.4	.3	.2	.1	.0	TH2
E1 _H	T2PS1	-	-	-	CTF	CLK2	CLK1	CLK0	CTCON 3)
E1 _H	T2PS1	-	ICR	ICS	CTF	CLK2	CLK1	CLK0	CTCON 4)
DE _H	.7	.6	.5	.4	.3	.2	.1	.0	CTRELL 3) 4)
DF _H	.7	.6	.5	.4	.3	.2	.1	.0	CTRELLH 3) 4)
F7 _H	.7	.6	.5	.4	.3	.2	.1	.0	CMSEL 3) 4)
C1 _H	COCA H3	COCA L3	COCA H2	COCA L2	COCA H1	COCA L1	COCA H0	COCA L0	CCEN
C9 _H	-	COCO N2	COCO N1	COCO N0	COCO EN0	COCA H4	COCA L4	COMO	CC4EN 3)
C9 _H	COCO EN1	COCO N2	COCO N1	COCO N0	COCO EN0	COCA H4	COCA L4	COMO	CC4EN 4)
F6 _H	.7	.6	.5	.4	.3	.2	.1	.0	CMEN 3) 4)
CA _H	.7	.6	.5	.4	.3	.2	.1	.0	CRCL
CB _H	.7	.6	.5	.4	.3	.2	.1	.0	CRCH
A1 _H	.7	.6	.5	.4	.3	.2	.1	.0	COMSETL 4)
A2 _H	.7	.6	.5	.4	.3	.2	.1	.0	COMSETH 4)
A3 _H	.7	.6	.5	.4	.3	.2	.1	.0	COMCLRL 4)
A4 _H	.7	.6	.5	.4	.3	.2	.1	.0	COMCLRH 4)
A5 _H	.7	.6	.5	.4	.3	.2	.1	.0	SETMSK 4)
A6 _H	.7	.6	.5	.4	.3	.2	.1	.0	CLRMSK 4)
C2 _H	.7	.6	.5	.4	.3	.2	.1	.0	CCL1
C3 _H	.7	.6	.5	.4	.3	.2	.1	.0	CCH1
C4 _H	.7	.6	.5	.4	.3	.2	.1	.0	CCL2
C5 _H	.7	.6	.5	.4	.3	.2	.1	.0	CCH2
C6 _H	.7	.6	.5	.4	.3	.2	.1	.0	CCL3
C7 _H	.7	.6	.5	.4	.3	.2	.1	.0	CCH3
CE _H	.7	.6	.5	.4	.3	.2	.1	.0	CCL4 3) 4)
CF _H	.7	.6	.5	.4	.3	.2	.1	.0	CCH4 3) 4)
D2 _H	.7	.6	.5	.4	.3	.2	.1	.0	CMLO 3) 4)
D3 _H	.7	.6	.5	.4	.3	.2	.1	.0	CMH0 3) 4)
D4 _H	.7	.6	.5	.4	.3	.2	.1	.0	CML1 3) 4)
D5 _H	.7	.6	.5	.4	.3	.2	.1	.0	CMH1 3) 4)

E1 _H	T2PS1	-	ICR	ICS	CTF	CLK2	CLK1	CLK0	CTCON
									4)

Fail Save Mechanisms

A8 _H	EAL	WDT	ET2	ES	ET1	EX1	ET0	EX0	IEN0
									1) 2)

A8 _H	EAL	WDT	ET2	ES0	ET1	EX1	ET0	EX0	IEN0
									3) 4)

B8 _H	EX EN2	SWDT	EX6	EX5	EX4	EX3	EX2	EADC	IEN1

A9 _H	-	WDTS	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	IP0
									1)

A9 _H	OWDS	WDTS	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	IP0
									2) 3) 4)

86 _H	.7	.6	.5	.4	.3	.2	.1	.0	WDTREL
									2) 3) 4)

Power Saving Modes

87 _H	SMOD	PDS	IDLS	-	GF1	GF0	PDE	IDLE	PCON
									1) 2)

87 _H	SMOD	PDS	IDLS	SD	GF1	GF0	PDE	IDLE	PCON
									3) 4)


- 1) SAB 80C515 / 80C535
- 2) SAB 80C515A / 83C515A-5
- 3) SAB 80C517 / 80C537
- 4) SAB 80C517A / 83C517A-5

--	--	--	--	--	--	--	--	--	--

Bit and byte addressable SFR

--	--	--	--	--	--	--	--	--	--

Not bit addressable SFR

 The shaded bits of an SFR are not used in the corresponding functional SFR group

B0 _H	RD	WR	T1	T0	INT1	INT0	TxD	RxD	P3
									1) 2)

B0 _H	RD	WR	T1	T0	INT1	INT0	TxD0	RxD0	P3
									3) 4)

E8 _H	.7	.6	.5	.4	.3	.2	.1	.0	P4

F8 _H	.7	.6	.5	.4	.3	.2	.1	.0	P5

DB _H	.7	.6	.5	.4	.3	.2	.1	.0	P6
									1) 2)

FA _H	.7	.6	.5	.4	.3	.2	.1	.0	P6
									3) 4)

DB _H	.7	.6	.5	.4	.3	.2	.1	.0	P7
									3) 4)

DD _H	-	-	-	-	.3	.2	.1	.0	P8
									3) 4)

D6 _H	.7	.6	.5	.4	.3	.2	.1	.0	CML2
									3) 4)

D7 _H	.7	.6	.5	.4	.3	.2	.1	.0	CMH2
									3) 4)

E2 _H	.7	.6	.5	.4	.3	.2	.1	.0	CML3
									3) 4)

E3 _H	.7	.6	.5	.4	.3	.2	.1	.0	CMH3
									3) 4)

E4 _H	.7	.6	.5	.4	.3	.2	.1	.0	CML4
									3) 4)

E5 _H	.7	.6	.5	.4	.3	.2	.1	.0	CMH4
									3) 4)

E6 _H	.7	.6	.5	.4	.3	.2	.1	.0	CML5
									3) 4)

E7 _H	.7	.6	.5	.4	.3	.2	.1	.0	CMH5
									3) 4)

F2 _H	.7	.6	.5	.4	.3	.2	.1	.0	CML6
									3) 4)

F3 _H	.7	.6	.5	.4	.3	.2	.1	.0	CMH6
									3) 4)

F4 _H	.7	.6	.5	.4	.3	.2	.1	.0	CML7
									3) 4)

F5 _H	.7	.6	.5	.4	.3	.2	.1	.0	CMH7
									3) 4)

SAB 80C5XX Microcontroller Family Pocket Guide	
Revision History :	08.98
Previous Releases :	08.96
Item	Subjects (changes since last revision)
Instruction List (long version)	MOV dadr,Rr : binary encoding corrected Several writing errors in columns "Function" and "Funktionsbeschreibung" corrected
Instruction List in Hexadecimal Ord.	Mnemonics for Hexcodes 75 _H to 7F _H and AD _H to AF _H corrected