

Befehlssatz für das Prozessormodell: "Mini-Power-PC"

Maschinen-Code (Op-Code)	Mnemonics ("Assembler")	Kurzbeschreibung	Beschreibung
0 0 0 0 x x 1 0 1 < n o t u >	CLR <i>Rnr</i>	« <i>Rxx</i> » := 0	Lösche das Register « <i>Rxx</i> » (alle Bit auf 0 setzen) und das Carry-Flag; 00 bis 11 für: Akku, R-1, R-2 bzw. R-3).
0 0 0 0 x x 1 1 1 < n o t u >	ADD <i>Rnr</i>	<i>Akku</i> := <i>Akku</i> + « <i>Rxx</i> »	Addition zweier 16-Bit-Zahlen (Zahl im Akku und Zahl im Register « <i>Rxx</i> »; 00 bis 11 für Akku, R-1, R-2 bzw. R-3) im 2er-Komplement; bei Überlauf wird das Carry-Flag gesetzt (= 1), sonst auf den Wert 0.
1 - - - - Z a h l - - - - >	ADDD # <i>Zahl</i>	<i>Akku</i> := <i>Akku</i> + # <i>Zahl</i>	Addition der 16-Bit-Zahl im Akku mit der 15-Bit-Zahl als direkten Operanden im 2er-Komplement; bei Überlauf wird das Carry-Flag gesetzt (= 1), sonst auf den Wert 0. Vor der Addition wird die 15-Bit-Zahl des Operanden auf 16 Bit erweitert (mit MSb des MSB auf 1 wenn negativ, sonst auf 0).
0 0 0 0 0 0 0 1 n o t u s e d	INC	<i>Akku</i> := <i>Akku</i> + 1	Der Akku (16-Bit-Zahl im 2er-Komplement) wird um den Wert 1 inkrementiert; bei Überlauf wird das Carry-Flag gesetzt (= 1), sonst auf den Wert 0.
0 0 0 0 0 1 0 0 n o t u s e d	DEC	<i>Akku</i> := <i>Akku</i> - 1	Der Akku (16-Bit-Zahl im 2er-Komplement) wird um den Wert 1 dekrementiert; bei Überlauf wird das Carry-Flag gesetzt (= 1), sonst auf den Wert 0.
0 1 0 - x x < A d r e s s e - >	LWDD <i>Rnr</i> , # <i>Adr</i>	« <i>Rnr</i> » := Inhalt Speicher(<i>Adr</i>)	In das Register mit der Nummer <i>xx</i> (00 bis 11 für Akku, R-1, R-2 bzw. R-3) wird der Inhalt der Speicherzellen <i>Adr</i> und <i>Adr</i> + 1 (1 Wort = 2 Byte) geladen. Mit 10 Bit können 1KiB Speicher adressiert werden.
0 1 1 - x x < A d r e s s e - >	SWDD <i>Rnr</i> , # <i>Adr</i>	Inhalt Speicher(<i>Adr</i>) := « <i>Rnr</i> »	In die über <i>Adr</i> und <i>Adr</i> + 1 adressierten Speicherzellen wird der Inhalt des Registers <i>xx</i> (00 bis 11 für Akku, R-1, R-2 bzw. R-3) geschrieben. Mit 10 Bit können 1KiB Speicher adressiert werden.
0 0 0 0 0 1 0 1 n o t u s e d	SRA	<i>Akku</i> := <i>Akku</i> / 2	Schieben <i>arithmetisch</i> nach rechts: der Inhalt des Akkus wird um eine Stelle nach rechts geschoben; der Inhalt vom LSB des LSB (das rechte Bit des Wortes) wird als Carry-Flag gesetzt. Dabei bleibt das MSb des MSB (Vorzeichenbit) und das 2. Bit des MSB erhalten.
0 0 0 0 1 0 0 0 n o t u s e d	SLA	<i>Akku</i> := <i>Akku</i> x 2	Schieben <i>arithmetisch</i> nach links: der Inhalt des Akkus wird um eine Stelle nach links geschoben; der Inhalt vom 2. Bit des MSB (das 2. Bit des Wortes) wird als Carry-Flag gesetzt. Dabei bleibt das MSb des MSB (Vorzeichenbit) erhalten. In das LSB des LSB (das letzte Bit des Wortes) wird eine 0 geschrieben.
0 0 0 0 1 0 0 1 n o t u s e d	SRL	<i>Akku</i> := <i>Akku</i> / 2	Schieben <i>logisch</i> nach rechts: der Inhalt des Akkus wird um eine Stelle nach rechts geschoben; der Inhalt vom LSB des LSB (das rechte Bit des Wortes) wird als Carry-Flag gesetzt. Das MSb des MSB (das 1. Bit des Wortes) wird auf 0 gesetzt.
0 0 0 0 1 1 0 0 n o t u s e d	SLL	<i>Akku</i> := <i>Akku</i> x 2	Schieben <i>logisch</i> nach links: der Inhalt des Akkus wird um eine Stelle nach links geschoben; der Inhalt vom LSB des LSB (das rechte Bit des Wortes) wird mit 0 aufgefüllt, das MSb des MSB (das 1. Bit des Wortes) wird als Carry-Flag gesetzt.
0 0 0 0 x x 1 0 0 < n o t u >	AND <i>Rnr</i>	<i>Akku</i> := <i>Akku</i> AND « <i>Rxx</i> »	Akku und Register <i>xx</i> (00 bis 11 für Akku, R-1, R-2 bzw. R-3) werden bitweise logisch mit AND verknüpft.
0 0 0 0 x x 1 1 0 < n o t u >	OR <i>Rnr</i>	<i>Akku</i> := <i>Akku</i> OR « <i>Rxx</i> »	Akku und Register <i>xx</i> (00 bis 11 für Akku, R-1, R-2 bzw. R-3) werden bitweise logisch mit ODER verknüpft.
0 0 0 0 0 0 0 0 < n o t u s >	NOT	<i>Akku</i> := NOT (<i>Akku</i>)	Alle Bit im Akku werden bitweise negiert.
0 0 0 1 x x 1 0 < n o t u s >	BZ <i>Rnr</i>	Bedingter Sprung	Wenn der Akku 0 ist, verzweige an die durch das Register <i>xx</i> (01 bis 11 für R-1, R-2 bzw. R-3) angegebene Speicheradresse; sonst wird der folgende Befehl normal fortgeführt.
0 0 0 1 x x 0 1 < n o t u s >	BNZ <i>Rnr</i>	Bedingter Sprung	Wenn der Akku ≠ 0 ist, verzweige an die durch das Register <i>xx</i> (01 bis 11 für R-1, R-2 bzw. R-3) angegebene Speicheradresse; sonst wird der folgende Befehl normal fortgeführt.
0 0 0 1 x x 1 1 < n o t u s >	BC <i>Rnr</i>	Bedingter Sprung, Carry	Wenn das Carry-Flag gesetzt ist (= 1), verzweige an die durch das Register <i>xx</i> (01 bis 11 für R-1, R-2 bzw. R-3) angegebene Speicheradresse, sonst wird der folgende Befehl normal fortgeführt.
0 0 0 1 x x 0 0 < n o t u s >	B <i>Rnr</i>	Unbedingter Sprung	Verzweige an die durch das Register <i>xx</i> (01 bis 11 für R-1, R-2 bzw. R-3) angegebene Speicheradresse.
0 0 1 1 0 - < A d r e s s e - >	BZD # <i>Adr</i>	Bedingter Sprung, direkt	Wenn der Akku 0 ist, verzweige an die durch den Operanden angegebene Speicheradresse; sonst wird das Programm mit dem folgenden Befehl fortgesetzt. Mit 10 Bit können 1KiB Speicher adressiert werden.
0 0 1 0 1 - < A d r e s s e - >	BNZD # <i>Adr</i>	Bedingter Sprung, direkt	Wenn der Akku ≠ 0 ist, verzweige an die durch den Operanden angegebene Speicheradresse; sonst wird das Programm mit dem folgenden Befehl fortgesetzt. Mit 10 Bit können 1KiB Speicher adressiert werden.
0 0 1 1 1 - < A d r e s s e - >	BCD # <i>Adr</i>	Bedingter Sprung, direkt	Wenn das Carry-Flag gesetzt ist, verzweige an die durch den Operanden angegebene Speicheradresse; sonst wird das Programm mit dem folgenden Befehl fortgesetzt. Mit 10 Bit können 1KiB Speicher adressiert werden.
0 0 1 0 0 - < A d r e s s e - >	BD # <i>Adr</i>	Unbedingter Sprung, direkt	Verzweige an die durch den Operanden angegebene Speicheradresse. Mit 10 Bit können 1KiB Speicher adressiert werden.