

Befehl	Argumente*	Beschreibung
add	-	Addiert die zwei obersten Elemente des Stacks und entfernt diese. Das Ergebnis wird auf den Stack gelegt.
and	-	Verknüpft die zwei obersten Elemente mit der AND-Operation und entfernt diese. Das Ergebnis wird auf den Stack gelegt.
dec	-	Dekrementiert den obersten Wert auf dem Stack.
inc	-	Inkrementiert den obersten Wert auf dem Stack.
inv	-	Invertiert den obersten Wert auf dem Stack.
or	-	Verknüpft die zwei obersten Elemente mit der OR-Operation und entfernt diese. Das Ergebnis wird auf den Stack gelegt.
nop	-	Keine Operation.
sub	-	Subtrahiert die zwei obersten Elemente des Stacks und entfernt diese. Das Ergebnis wird auf den Stack gelegt. Dabei wird das zweite Element „minus“ das oberste Element gerechnet.
swap	-	Vertauscht die zwei obersten Elemente auf dem Stack.
xor	-	Verknüpft die zwei obersten Elemente mit der XOR-Operation und entfernt diese. Das Ergebnis wird auf den Stack gelegt.
call	<addr> [14]	(Funktionsaufruf) Springt an die Stelle von <addr> und führt den dort stehenden Code aus, bis ein return erreicht wird. <b>Wichtig:</b> Damit „return“ wieder an die korrekte Stelle zurückspringt, muss der Stack am Ende der „Funktion“ im gleichen Zustand wie zu Beginn der „Funktion“ sein.
return	-	Springt zurück an die Stelle des letzten Funktionsaufrufes.
goto	<addr> [14]	Springt bedingungslos an die Stelle <addr>.
jnz	<addr> [14]	Springt an die Stelle <addr>, falls das oberste Element des Stacks gleich null ist.
jnz	<addr> [14]	Springt an die Stelle <addr>, falls das oberste Element des Stacks ungleich null ist.
jnc	<addr> [14]	Springt an die Stelle <addr>, falls die obersten zwei Stackelemente gleich sind.
pop	-	Entfernt das oberste Element vom Stack.
push	<elem> [8]	Legt den Wert <elem> auf den Stack.
shl8	-	Führt eine Schiebeoperation um 8 Bits nach links auf dem obersten Stackelement aus.
shr1	-	Führt eine Schiebeoperation um 1 Bit nach rechts auf dem obersten Stackelement aus.
dup	-	Erstellt eine Kopie des obersten Stackelements und legt diese auf den Stack.
dload	-	Lädt einen 8-Bit Wert aus dem RAM von der Adresse, welche an erster Stelle auf dem Stack liegt und ersetzt das oberste Stackelement mit dem Wert.
dstore	-	Speichert das zweite Element auf dem Stack an der Adresse, welche an erster Stelle auf dem Stack

		liegt. Nur die Adresse (1. Element) wird vom Stack entfernt, nicht der Wert (2. Element).
--	--	---

\* Die Zahl in den eckigen Klammern entspricht der Größe in Bit.

Befehl	OP Code (dezimal)	OP Code (hexadezimal)
add	128	80
and	131	83
dec	134	86
inc	136	88
inv	166	a6
or	138	8a
nop	0	0
sub	141	8d
swap	144	90
xor	150	96
call	156	9c
return	161	a2
goto	30	1e
jmz	37	25
jmnz	44	2c
jmc	52	34
pop	60	3c
push	65	41
shl8	70	46
shr1	73	49
dub	75	4b
dloads	96	60
dstore	105	69

### Speicheraufteilung:

Name	Anz. Zellen	Einheiten	Adressierungsbereich		Gesamtspeicheranteil
			virtuell	physisch	
STACK	256	32 Bit	0-255	0-255	25% (1024 Byte)
GPM	256	32 Bit	256-511	256-511	25% (1024 Byte)
PROGRAM	2048	8 Bit	2048-4095	512-1023	50% (2048 Byte)

