

Hack-Assembly - Symbole

Anweisungstypen

Anweisungstyp	Bedeutung
A-Anweisungen	address instructions
C-Anweisungen	compute instructions
L-Anweisungen	labels declaration instructions

Vordefinierte Symbole

Symbol	Beschreibung
A	A-Register (Adressenregister)
D	D-Register (Datenregister)
M	Hauptspeicher-Register (Adresse A)
SP	RAM-Adresse 0
LCL	RAM-Adresse 1
ARG	RAM-Adresse 2
THIS	RAM-Adresse 3
THAT	RAM-Adresse 4
R0-R15	RAM-Register (16)
SCREEN	16384 Adresse des Bildschirmspeichers
KBD	24576 Adresse des Tastaturregister

Hack-Assembly - A-Anweisungen

@value

SOMETHING

Verwendung der A-Anweisung

@value	Vorgang	Beschreibung
$D = A$	$D \leftarrow value$	Laden einer Konstante
$D = M$	$D \leftarrow RAM[value]$	Auswahl Datenspeicherzelle
JMP	$fetch\ ROM[value]$	Auswahl Befehlsspeicherzelle

Hack-Assembly - L-Anweisungen

(LABEL)

// Instructions

@LABEL

0; JMP

Die Anweisung (LABEL) deklariert ein neues Label mit dem Name 'Label'. Der Assembler übersetzt diese dann in die Adresse der nächsten Anweisung (nachfolgende Zeile)

Hack-Assembly - C-Anweisungen

DEST = **COMP** ; **JUMP**
 Speicherort des Ergebnisses Auszuführende Berechnung Sprungbedingung

Computation Symboltabelle (ALU)

x: D y: A $m = 1 \rightarrow A = M$							
zx	nx	zy	ny	f	no	$m = 0$	$m = 1$
1	0	1	0	1	0	0	
1	1	1	1	1	1	1	
1	1	1	0	1	0	-1	
0	0	1	1	0	0	D	
1	1	0	0	0	0	A	M
0	0	1	1	0	1	$D/!D$	
1	1	0	0	0	1	$A/!A$	$M/!M$
0	0	1	1	1	1	$-D$	
1	1	0	0	1	1	$-A$	$-M$
0	1	1	1	1	1	$D + 1$	
1	1	0	1	1	1	$A + 1$	$M + 1$
0	0	1	1	1	0	$D - 1$	
1	1	0	0	1	0	$A - 1$	
0	0	0	0	1	0	$D + A$	$D + M$
0	1	0	0	1	1	$D - A$	$D - M$
0	0	0	1	1	1	$A - D$	$M - D$
0	0	0	0	0	0	$D \& A$	$D \& M$
0	1	0	1	0	1	$D A$	$D M$

Destination Symboltabelle

M: Memory[A] D: D-Register A: A-Register				
Symbol	d_1	d_2	d_3	Speicherort
-	0	0	0	Keine Speicherung
M	0	0	1	Memory[A]
D	0	1	0	D Register
MD	0	1	1	Memory[A] & D-Register
A	1	0	0	A-Register
AM	1	0	1	A-Register & Memory[A]
AD	1	1	0	A-Register und D-Register
AMD	1	1	1	A-Reg., Memory[A] & D-Reg.

Jump Symboltabelle

E: EQUAL G: GREATER L: LOWER				
Symbol	j_2	j_1	j_0	Sprungbedingung
-	0	0	0	Sprung niemals
JGT	0	0	1	Sprung falls $out > 0$
JEQ	0	1	0	Sprung falls $out = 0$
JGE	0	1	1	Sprung falls $out \geq 0$
JLT	1	0	0	Sprung falls $out < 0$
JNE	1	0	1	Sprung falls $out \neq 0$
JLE	1	1	0	Sprung falls $out \leq 0$
JMP	1	1	1	Sprung immer