

# Software

There isn't any. You'll have to write your own.

The first instruction executed is always a HOP 0. SGCS generates this instruction for you automatically; you don't need to code it. You must put a HOP word at address 0, sector 0 to tell SGSC what to do next.

Here's a short example. The memory address is shown in a column on the left side. 'X' means we don't care whether the bit is zero or one.

Addresses 0, 5, 6, and 7 contain 26-bit words: HOP words at 0 and 7; data words at 5 and 6.

Execution begins by processing the HOP word at address 0. The word says "begin execution at syllable 0 (bit 15=0), sector 0 (bits 13-9), address 1 (bits 8-1)". Execution therefore begins at the instruction in syllable 0 (bits 13-1) of memory address 1.

The instruction is a Clear and Add (CLA) with the operand pointing to the data word at address 5 which contains the integer "3". Number 3 is loaded into the accumulator.

The next instruction is ADD 6. It adds the data word at address 6 (-1 in 2's compliment representation) to the accumulator; this decrements the accumulator.

The next instruction, TNZ 2, jumps to the operand address if the accumulator is nonzero (it is). SGSC jumps back to the ADD 6 instruction and decrements the accumulator again.

When the accumulator reaches zero, the TNZ instruction falls through to the next instruction: HOP 7. This executes the HOP word at address 7 which transfers execution to sector 0, word 1, syllable 1. The instruction there is TRA 1 which causes SGSC to infinitely loop.

ADDR	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01		
0	X	X	X	X	X	X	X	X	X	X	X	0	X	0	0	0	0	0	0	0	0	0	0	0	0	0	1	HOP WRD
1	TRA 1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0	1	CLA 5
2														0	1	0	0	0	0	0	0	0	0	1	1	0		ADD 6
3														1	1	1	1	X	0	0	0	0	0	0	1	0		TNZ 2
4														0	0	0	0	0	0	0	0	0	0	1	1	1		HOP 7
5		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	DATA WRD
6		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	DATA WRD
7	X	X	X	X	X	X	X	X	X	X	X	X	1	X	0	0	0	0	0	0	0	0	0	0	0	0	1	HOP WRD