6.0	Appendix	6 - 1
6.1	Cycles 8T/Sprint 8T	6 - 1
6.2	Drilling cycles 8M/8MC/Sprint 8M	6 - 16
6.3	Drilling and milling patterns Sprint 8M	6 - 37
6.4	Drilling and milling patterns 8M/8MC	6 - 53
6.5	Address codes for the address parameters	6 - 69
6.6	Figure for stock removal cycle on page 3.3.1	6 - 70
6.7	Figure for stock removal cycle on page 3.3.2	6 - 71
6.8	Defined R parameters	6 - 72
6.9	Overview of store access (Values)	6 - 73
6.9.1	SINUMERIK 8T/Sprint 8T	6 - 73
6.9.2	SINUMERIK Sprint 8M	6 - 74
6.9.3	SINUMERIK 8M/8MC	6 - 75
6.9.4	SINUMERIK 8N	6 - 76

The controls are capable of their respective functions as described.
This does not imply that the functions were fitted at delivery or if they are available for service use.

We reserve the right to amend or change this publication.

0.0 Overview of "@" and "&" functions

	Function	8T	Sprint 8T	8M	Sprint 8M	8MC	8N	Chapter
@ 00	Unconditional jump	(X)	X	(X)	X	X	X	1.1
@ 01	Conditional jump - equals	(X)	X	(X)	X	X	X	1.2
@ 02	Conditional jump - larger	(X)	X	(X)	X	X	X	1.2
@ 03	Conditional jump - larger/equal	(X)	X	(X)	X	X	X	1.2
@ 10	Square root	(X)	X	(X)	X	X	X	2.1
@ 15	Sine	(X)	X	(X)	X	X	X	2.2
@ 18	Arcustangens *							2.3
@ 20	Load address parameter	(X)	X	(X)	X	X	X	3.1
@ 21	Reference preparation	(X)	X	-	-	-	-	3.2
@ 22	Cutter point calculation	(X)	X	-	-	-	-	3.2
@ 23	Tool change	(X)	X	-	-	-	-	3.4
@ 24	Load actual value	(X)	X	(X)	X	X	X	3.5
@ 25	Start conditions for cycles	(X)	X	(X)	X	X	X	3.6
@ 29	Read/Load System stores	(X)	X	(X)	X	X	X	3.7
@ 30	Reference to machine actual value	(X)	X	(X)	X	X	X	4.2
@ 31	Clear buffer	(X)	X	(X)	X	X	-	4.1
@ 90	Address parameter	(X)	X	(X)	X	X	X	5.1
@ 91	Address parameter	(X)	X	(X)	X	X	X	5.1
@ 92	Address parameter	(X)	X	(X)	X	X	X	5.1
@ 93	Address parameter	(X)	X	(X)	X	X	X	5.1
@ 94	Address parameter	(X)	X	(X)	X	X	X	5.1
@ 95	Address parameter	(X)	X	(X)	X	X	X	5.1
@ 96	Address parameter	(X)	X	(X)	X	X	X	5.1
@ 97	Address parameter	(X)	X	(X)	X	X	X	5.1
@ 98	Address parameter	(X)	X	(X)	X	X	X	5.1
@ 99	Address parameter	(X)	X	(X)	X	X	X	5.1
&	Operator Aids	(X)	(X)	(X)	(X)	(X)	(X)	5.2

X Available

(X) Available, but cannot be entered from the operator's panel

* From Software 02 onwards

0.1 Introduction

This description covers all System 8 controls. The System 8 cycles are not part of the hardware but stored in the control's program memory. In order to realise the powerful System 8 cycles, it was necessary to develop a number of functions which were developed on from System 7. These functions are called up with the @ address. The purpose of this description is to describe and clarify the functions with examples, thus enabling the customer to program machine and technologically orientated cycles.

In order to understand this description a study of the programming manual 8T/Sprint 8T or 8M/Sprint 8M/8MC or 8N is a pre-requisite.

Parameter Chaining

Calculation Type	Programmed Calculation	Argument	Result found in
Addition	R01 R02	R01+R02	R01
Subtraction	R01-R02	R01-R02	R01
Multiplication	R01.R02	R01.R02	R01
Division	R01/R02	R01:R02	R01
Definition +	R01 10 R02	R01 +10	
Addition		R01+R02	R01
Definition -	R01-10-R02	R01-10	
Subtraction		R01-R02	R01

Note:
Each chain operation must be programmed as a separate block.

General

Programmed parameter chaining and programmed @ functions initiate calculation times which can be up to 10 ms per link. It is therefore necessary to ensure that the calculations be programmed approximately 10 blocks before required. If e.g. a move is executed at a suitable speed, the control can be allowed sufficient time to complete the next 10 calculations. The signal "cycles lock" also enables faster calculations (see chapter 2.2.1 section g).

In conjunction with an @ program definition, it is often necessary to transfer a parameter via the interface. See 3.5.1 (Loading the position value).

R parameter range:

a) Programmable and displayable range

Largest value: ± 99999999.

Smallest value: ± .00000001

1.1 @ 00 "Unconditional jump"

Application: With a conditional (absolute) jump, it is possible to jump over parts of a program. The jumped blocks are not executed.

Example: See 1.2.1; 2.2.1

Programming: @ 00 ± 1234

Command: Unconditional jump

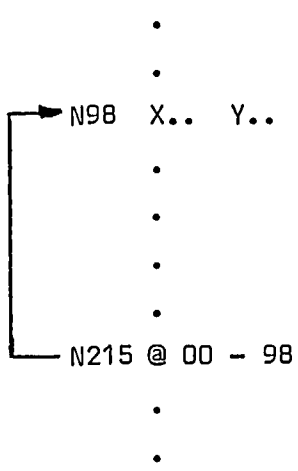
The jump target precedes (-) or follows (+) the jump command. The control searches for the jump target in the defined direction.

Block number of the jump target max. 4 decades

For a special case, an R parameter and its sign can be added in order to generate different jump targets. This special case is shown on the right hand side of the following figures.

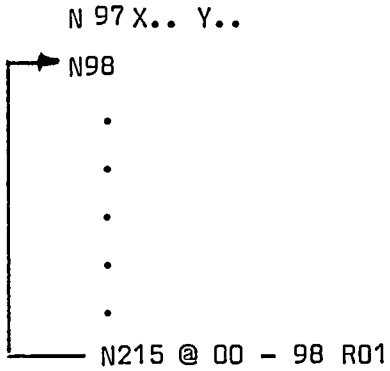
The jump target precedes the jump command (-)

Normal case

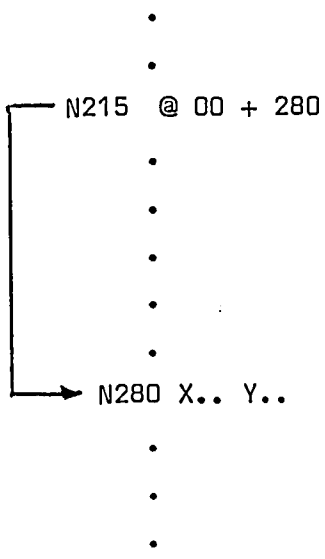


Special case

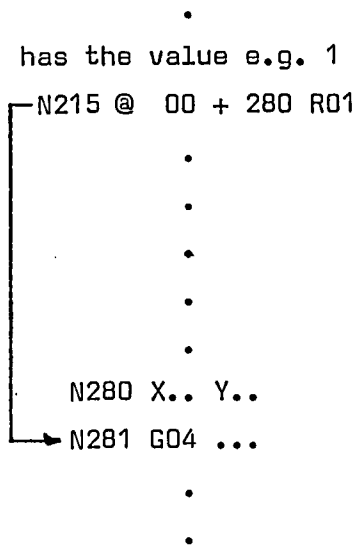
R01 has the value e.g. 1



The jump target follows the jump command (+)



R01 has the value e.g. 1



Note: Jump targets must always be blocks <u>with</u> block numbers. This also applies when the jump target is changed by an R parameter.

A jump requires time (max 10 msec per jumped block)

1.2 @ 01, @ 02, @ 03 "Conditional jump"

Application: The conditional jumps permit program branching dependent upon the condition:
equal to @ 01
larger than @ 02
larger than or equal to @ 03

Example: See 1.2.1; 2.2.1

Programming: @.. ± 1234 R.. R..

Command and jump condition

The jump target precedes (-) or follows the jump command. The control searches for the jump target in the defined direction

Block number of the jump target, max 4 decades

R-parameter comparison for the conditional jump:
1st parameter equal to the 2nd parameter @ 01
1st parameter larger than the 2nd parameter @ 02
1st parameter larger than or equal to the 2nd parameter @ 03

1st parameter

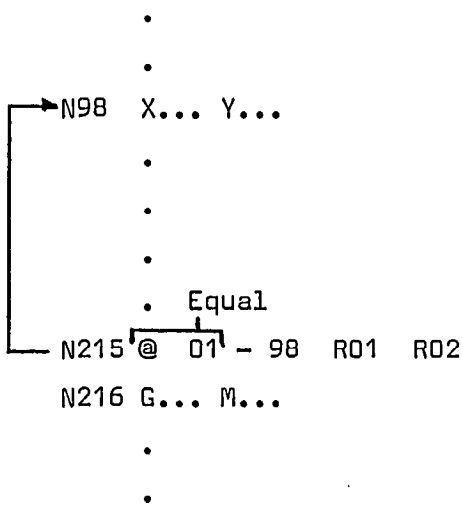
The control compares the parameter contents in the mathematical sense
(e.g. - 213 is smaller than - 7)

2nd parameter

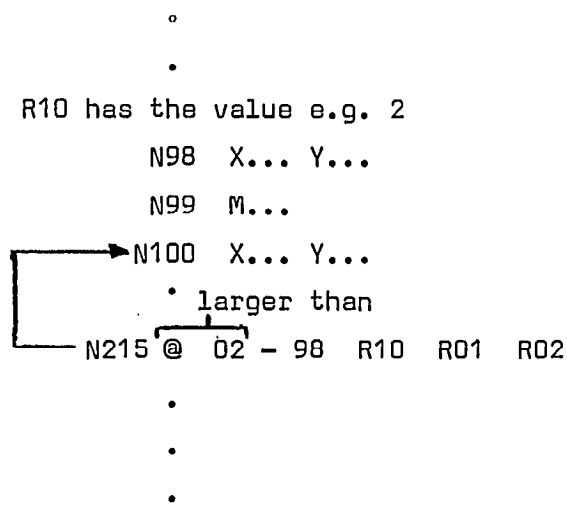
For a special case, an R parameter and its sign can be added in order to generate different jump targets. This special case is shown on the right hand side of the following figures.

The jump target precedes the jump command (-)

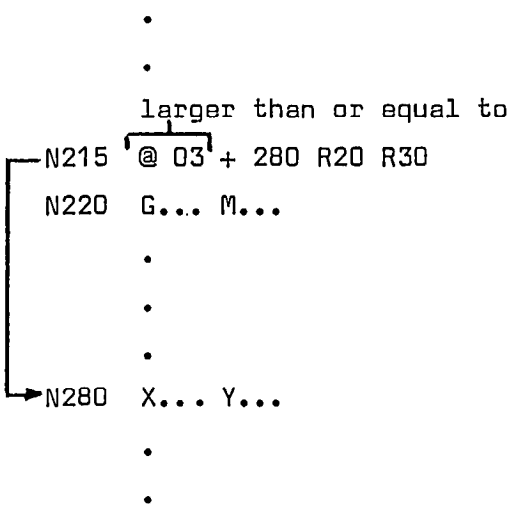
Normal case

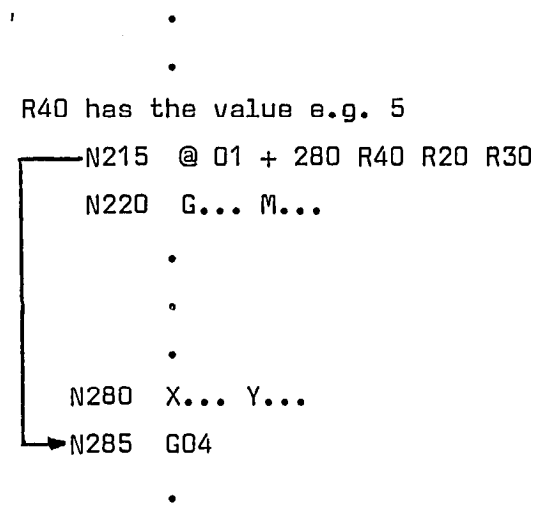


Special case



The jump target follows the jump command (+)





Note: Jump targets must always be blocks <u>with</u> a block number. This also applies when the jump target is changed by an R parameter.

A jump requires time. (max 10 msec per jumped block)

1.2.1 Summary example of conditional and unconditional jumps

Task definition: Within a program there must be a branch to another program

Flow chart:

