



Zur Verarbeitung von Daten ist es notwendig, diese Daten mittels Operatoren (kurz OZ für Operatorzeichen) miteinander zu verknüpfen. Operatoren werden hinsichtlich der Zahl ihrer Operanden (kurz OP) unterschieden (unär – ein Operand / binär – zwei Operanden / ternär – drei Operanden).

	Pri.	Ass.	OZ	Syntax	Beschreibung	
mathematische O.	12	L	+ -	Erg = OP ₁ + OP ₂	Summe, Differenz	Arithmetische Operatoren werden auf numerische Werte angewendet. Hierbei ist der Datentyp des Ergebnisses Erg abhängig von den Datentypen der Operanden OP ₁ und OP ₂ .
	13	L	* /	Erg = OP ₁ * OP ₂	Multiplikation, Division	
	13	L	%	Erg = OP ₁ % OP ₂	Divisionsrest	
	14	R	++ --	Erg = ++OP	Pre-Inkrement Pre-Dekrement	
	15	L	++ --	Erg = OP++	Post-Inkrement Post-Dekrement	

	Pri.	Ass.	OZ	Syntax	Beschreibung	
Zuwei-	2	R	=	Erg = OP	Wertzuweisung	Auch hier ist der Datentyp des Ergebnisses abhängig vom Datentyp des Operanden. x+= y entspricht: x = x+y
	2	R	+= -=	Erg += OP (auch *=, /=, %=)	Operation mit anschließender Wertzuweisung.	

	Pri.	Ass.	OZ	Syntax	Beschreibung	
Vergleichsoperato-	9	L	==	Erg = OP ₁ == OP ₂	Gleichheit	Vergleichsoperatoren können auf numerische Werte angewendet werden. Wird das boolesche Ergebnis gespeichert, so findet eine Konvertierung in den Datentyp bool statt. (für <code>int false=0</code> sonst <code>true</code>)
	9	L	!=	Erg = OP ₁ != OP ₂	Ungleichheit	
	10	L	< <=	Erg = OP ₁ < OP ₂	kleiner als, kleiner gleich als	
	10	L	> >=	Erg = OP ₁ >= OP ₂	größer als, größer gleich als	

	Pri.	Ass.	OZ	Syntax	Beschreibung	
logische O.	5	L	&&	Erg = OP ₁ && OP ₂	Logisches UND	Repräsentiert in Abhängigkeit der Bedingung (Bed) OP ₁ oder OP ₂
	4	L		Erg = OP ₁ OP ₂	Logisches ODER	
	14	R	!	Erg = ! OP	Logische Verneinung	
	3	L	? :	Bed ? OP ₁ : OP ₂	Bedingungsoperator	

	Pri.	Ass.	OZ	Syntax	Beschreibung	
Bitoperatoren	11	L	<<	Erg = OP << n	n-mal Links schieben	Mit Bit-Operationen wird auf die Binärdarstellung der Zahlen zugegriffen. Diese Operationen sind für Fließkomma-Operanden nicht zulässig.
	11	L	>>	Erg = OP >> n	n-mal Rechts schieben	
	8	L	&	Erg = OP ₁ & OP ₂	bitweise UND	
	7	L	^	Erg = OP ₁ ^ OP ₂	bitweise EXCLUSIV-ODER	
	6	L		Erg = OP ₁ OP ₂	bitweise ODER	
	14	R	~	Erg = ~ OP	bitweise Komplement	

Manipulation von einzelnen Bits in einem Datenwort (DW)		
Setzen des n-ten Bits im DW	Rücksetzen des n-ten Bits im DW	Invertieren des n-ten Bits im DW
DW = DW (1 << n)	DW = DW & ~ (1 << n)	DW = DW ^ (1 << n)

Priorisierung: Operatoren höherer Priorisierung werden vor Operatoren niedrigerer Priorisierung ausgewertet (z.B. **UND &&** Pri.5 geht vor **ODER ||** Pri.4).

Assoziativität: gibt an, ob eine Folge von Operatoren gleichen Vorrangs von links (L) oder von rechts (R) abgearbeitet wird.

