Transferoperationen und Operandenzugriff

HDN		Beispiel HDN	Bedeutung
<---		R1 <--R2	**Transfer, logisch** R2 wird nach R1 gemäß Länge (R1) übertragen
M		R1 <--M[x]	**Speicherzugriff** Wenn R1 32 Bit breit, dann werden bei Byteadressierung die Bytes M[x], M[x+1], M[x+2],M[x+3] übertragen
$<--_n$		M[y] $<--_{16}$M[x]	**Transfer mit expliziter Längenangabe** Übertragung von 16 Bits von Platz x nach Platz y
X_n		$R1_0$ <--0	**Zugriff zum Einzelbit** Löschen des Sign-Bits, d.h.MSB (Most Significant Bit) (Linksnumerierung)
$X_{m..n}$		$R3_{24..31}$<--M[x]	**Zugriff auf eine Bitkette** Übertrage Byte M[x] in das niederwertigste Byte von R3
X^m		$R3_{0..23}$<-- 0^{24}	**Wiederholungsspezifikation** die oberen 3 Byte von R3 werden auf "Null" gesetzt

HDN-Operatoren

HDN		Beispiel HDN	Bedeutung
##		R3 <-- 0^{24}##M[x] F2##F3$<--_{64}$M[x]	**Verkettungsoperator** **rechte Seite** M[x] im niederen Byte und obere 24 Bit = 0, **linke Seite** es werden 64 Bit vom Speicher zu dem Gleitkomma-Doppelregister F2/F3 übertragen
< <		R1 << 5	**Verschiebeoperator** Verschiebe R1 logisch nach links um 5Bit
> >		R1 >> 8	Verschiebe R1 logisch nach rechts um 8 Bit
$> >_a$		R1 $>>_a$ 16	Verschiebe R1 arithmetisch nach rechts um 16 Bit
= = != < >>>= <=	EQL NEQ LSS GTR GEQ LEQ	(R1 = = R2)&(R3 != R4)	**Relationale Operatoren** Ausdruck relationaler Operatoren ist "True", wenn die Relation "R1 gleich R2 "und die Relation" R3 ungleich R4" "True" ist
& | ^ !	AND OR XOR NOT	(R1 & (R2 | R3))	**Logische Operatoren** bitweise ODER-Verknüpfung von R2 mit R3; das Ergebnis wird UND-verknüpft mit R1
+ - * /		(R1+R2)/R3	**arithmetische Operatoren** Vorrangregeln wie in der HLL C