

ISL Opcodes

OPCODE	PARAMETRE	STAK	NOTER
<i>Symbolsk navn</i>	<i>Navn (datatype)</i>	<i>Navn (datatype) [pop] Første i listen = stak top</i>	<i>Beskrivelse</i>
PUSH	Datatype (tDataType) Value (= Datatype)	-	Lægger Value på stakken.
PUSHVAR	Datatype (tDataType) VarAddress (u.long)	-	Lægger indholdet af variabel på "VarAddress" på stakken
PUSHENGINEVAR	-	VarID (u.int) pop ObjID (u.int) pop	Lægger indholdet af engine variabel "ObjID.VarID" på stakken.
PUSHRETVAL	Datatype (tDataType)	-	Lægger en nulværdi af type "Datatype" på stakken, til brug som returværdi i funktioner.
PUSHREG	RegisterID	-	Lægger værdien af et register på stakken. Kan være regobject, regevent, regparameter1, regparameter2, regparameter 3 eller regparameter4.
PUSHFLAG	Address	-	Lægger værdien af flaget på Address på stakken (som boolean).
POPVAR ⁱ	Datatype (tDataType) VarAddress (u.long)	Value (?) pop	Lægger værdien "Value" i variabelen på "VarAddress".
POPENGINEVAR ⁱ	-	VarID (u.int) pop ObjID (u.int) pop Value (?) pop	Lægger værdien "Value" i engine variabel "ObjID.VarID".
POPN	NoOfElements (int)	-	Fjerner "NoOfElements" elementer fra stakken.
GETSTACKVAR	Index (int)	-	Lægger værdien af den angivne stak variabel øverst på stakken (Index 1 = øverste).
SETRETVAL	-	Value (?) pop	Sæt værdien af det aktuelle "Return value" stak elementet til "Value".
SETFLAG	Address	-	Sætter flaget på "Address" til "1".
CLEARFLAG	Address	-	Sætter flaget på "Address" til "0".
ADD ⁱⁱ	-	Operand 2 (?) pop Operand 1 (?) pop	Lægger resultatet af "Operand 1 + Operand 2" på stakken.
SUB ⁱⁱ	-	Operand 2 (?) pop Operand 1 (?) pop	Lægger resultatet af "Operand 1 - Operand 2" på stakken.
MUL ⁱⁱ	-	Operand 2 (?) pop Operand 1 (?) pop	Lægger resultatet af "Operand 1 * Operand 2" på stakken.
DIV ⁱⁱ	-	Operand 2 (?) pop Operand 1 (?) pop	Lægger resultatet af "Operand 1 / Operand 2" på stakken.

MOD ⁱⁱ	-	Operand 2 (?) pop Operand 1 (?) pop	Lægger resultatet af "Operand 1 MOD Operand 2" på stakken.
AND ⁱⁱⁱ	-	Operand 2 (?) pop Operand 1 (?) pop	Lægger resultatet af "Operand 1 AND Operand 2" på stakken.
OR ⁱⁱⁱ	-	Operand 2 (?) pop Operand 1 (?) pop	Lægger resultatet af "Operand 1 OR Operand 2" på stakken.
XOR ⁱⁱⁱ	-	Operand 2 (?) pop Operand 1 (?) pop	Lægger resultatet af "Operand 1 XOR Operand 2" på stakken.
CMPLT ^{iv}	-	Operand 2 (?) pop Operand 1 (?) pop	Lægger resultatet af "Operand 1 < Operand 2" på stakken.
CMPGT ^{iv}	-	Operand 2 (?) pop Operand 1 (?) pop	Lægger resultatet af "Operand 1 > Operand 2" på stakken.
CMPLTEQ ^{iv}	-	Operand 2 (?) pop Operand 1 (?) pop	Lægger resultatet af "Operand 1 <= Operand 2" på stakken.
CMPGTEQ ^{iv}	-	Operand 2 (?) pop Operand 1 (?) pop	Lægger resultatet af "Operand 1 >= Operand 2" på stakken.
CMPNEQ ^{iv}	-	Operand 2 (?) pop Operand 1 (?) pop	Lægger resultatet af "Operand 1 <> Operand 2" på stakken.
CMPEQ ^{iv}	-	Operand 2 (?) pop Operand 1 (?) pop	Lægger resultatet af "Operand 1 = Operand 2" på stakken.
NOT ^v	-	Operand 1 (?) pop	Lægger resultatet af "NOT Operand 1" på stakken.
NEG ^{vi}	-	Operand 1 (?) pop	Lægger resultatet af "- Operand 1" på stakken.
CALLFUNCTION	EntrypointAddress (u.long)	Parameter n (?) (= stakvar 1) ... Parameter 1 (?) Returværdi (?)	Lægger returadresse på stakken, sætter IP til "EntrypointAddress"
RETURNFUNCTION	-	ReturnAddress (u.long) pop	Sætter IP til "ReturnAddress".
CALLHANDLER	-	EventID (u.int) pop ObjID (u.int) pop Parameter 4 (?) (= stakvar 1) Parameter 3 (?) Parameter 2 (?) Parameter 1 (?)	Pop'er "EventID" og "ObjID", lægger returadresse på stakken, finder handlerliste, kalder alle handlere i liste, pop'er returadresse til IP.
RETURNHANDLER	Stop (int)	-	Afslutter "Execute" funktionen. Hvis "Stop" er 1 stoppes udførelsen af den aktuelle handlerliste ("stop" funktionen).
CALLDECISIONTABLE	EntrypointAddress (u.long)	EventID (u.int) pop ObjID (u.int) pop Parameter 4 (?) Parameter 3 (?) Parameter 2 (?) Parameter 1 (?)	Pop'er "EventID" og "ObjID", lægger returadresse på stakken, sætter IP til "EntrypointAddress".
RETURNDECISIONTABLE	-	ReturnAddress (u.long) pop	Sætter IP til "ReturnAddress".
SEQSTART	EntrypointAddress (u.long)	-	Lægger "EntrypointAddress" i listen over aktive sekvens tabeller, hvis den ikke allerede var der udføres tabellens rene

			"DO" statements ved at lægge returadressen på stakken. Derefter lægges "0" og derefter "1" på stakken (ved almindeligt kald lægges i stedet "0" og "0" på stakken).
SEQBREAK	EntrypointAddress (u.long)	-	Fjerner "EntrypointAddress" i listen over aktive sekvens tabeller. Hvis den var der udføres tabellens "WHEN BREAK" statements ved at lægge returadressen på stakken og derefter lægge "1" og derefter "0" på stakken.
SEQEND	EntrypointAddress (u.long)	-	Fjerner "EntrypointAddress" i listen over aktive sekvens tabeller.
RETURNSEQTABLE	-	ReturnAddress (u.long) pop	Sætter IP til "ReturnAddress".
JMP	Address (u.long)	-	Sætter IP til "Address".
JMPFALSE	Address (u.long)	Value (int) pop	Hvis "Value" er 0 sættes IP til "Address".
JMPTRUE	Address (u.long)	Value (int) pop	Hvis "Value" er ≥ 0 sættes IP til "Address".
JMPCASE	Address (u.long) Datatype (tDataType) CaseValue (= Datatype)	Value (?)	Hvis "CaseValue" er = "Value", sættes IP til "Address".
JMPCASEVAR	Address (u.long) Datatype (tDataType) CaseVarAddress (u.long)	Value (?)	Hvis variabelen på "CaseVarAddress" er = "Value", sættes IP til "Address".
INC	VarAddress (u.long) Value (int)	-	Lægges "Value" til indholdet af "VarAddress".
CHECKEVENT	-	EventID (u.int) pop ObjID (u.int) pop Parameter 4 (?) pop Parameter 3 (?) pop Parameter 2 (?) pop Parameter 1 (?) pop	Sammenligner "EventID" og "ObjID" med aktuelle decisiontable event- og objekt-ID'er. Sammenligner "Parameter 1 - 4" med "stak variable" parametrene (fra CALLDECISIONTABLE kaldet) på stakken. Hvis alle matcher (datatype Wildcard matcher alt) lægges "1" på stakken - ellers lægges "0".
EVALUATEEVENT	FlagAddress	EventID (u.int) pop ObjID (u.int) pop Parameter 4 (?) pop Parameter 3 (?) pop Parameter 2 (?) pop Parameter 1 (?) pop	Hvis flaget på "FlagAddress" ikke er sat, sammenlignes "EventID" og "ObjID" og "Parameter 1 - 4" med de tilsvarende registre. Hvis alle matcher (datatype Wildcard matcher alt) sættes flaget på "FlagAddress". Hvis flaget er sat udføres sammenligningen ikke. Flaget på "FlagAddress" lægges

			derefter på stakken (som ved PUSHFLAG).
--	--	--	---

ⁱ Typekompatibilitet ved tildeling (A := B):

A	B
Integer	Integer
-	Boolean
-	RandomInt
-	RandomIntSeq
-	IntSeq
Real	Real
-	Integer
-	Boolean
Boolean	Boolean
-	Integer
Timeout	Timeout
-	Integer
RandomInt	RandomInt
-	Range
RandomIntSeq	RandomIntSeq
-	Range
IntSeq	IntSeq
-	Range
Range	Range
-	RandomInt
-	RandomIntSeq
-	IntSeq

ⁱⁱ Typekompatibilitet ved dyadiske operatorer (A op B):**Tabel 1 (Add, Sub, Mul, Div)**

A	B	Resultat
Integer	Integer	Integer
-	Real	Real
Real	Real	Real
-	Integer	Real

Tabel 2 (Mod)

A	B	Resultat
Integer	Integer	Integer

ⁱⁱⁱ Typekompatibilitet ved dyadiske logiske operatorer (A op B):

A	B	Resultat
Integer	Integer	Boolean
-	Boolean	Boolean
-	Real	Boolean
Boolean	Boolean	Boolean
-	Integer	Boolean
Real	Real	Boolean
-	Integer	Boolean

^{iv} Typekompatibilitet ved dyadiske relationelle operatorer (A op B):**Tabel 1 (LT, GT, LTEQ, GTEQ)**

A	B	Resultat
----------	----------	-----------------

Integer	Integer	Boolean
-	Real	Boolean
-	Range	Boolean
Real	Real	Boolean
-	Integer	Boolean
-	Range	Boolean
Range	Integer	Boolean
-	Real	Boolean
-	Range	Boolean

Tabel 2 (NEQ, EQ)

A	B	Resultat
Integer	Integer	Boolean
-	Real	Boolean
-	Range	Boolean
Real	Real	Boolean
-	Integer	Boolean
-	Range	Boolean
Range	Integer	Boolean
-	Real	Boolean
-	Range	Boolean
Boolean	Boolean	Boolean

^v Typekompatibilitet ved monadiske logiske operatorer (op A):

A	Resultat
Integer	Boolean
Boolean	Boolean
Real	Boolean

^{vi} Typekompatibilitet ved monadiske operatorer (op A):

A	Resultat
Integer	Integer
Real	Real