Stack/Accumulator ISA with five PDP11 addressing modes **James C. Brakefield (2021)** Most instructions use two bytes, this sheet has the two byte op-codes

Op	# ops	description	std. stack operation	detailed descripton	comments
LD	0.5	Load zero extended from memory		Data size in 2nd byte	
LDS	0.5	Load sign extended from memory		Data size in 2nd byte	
LDF		Load floating-point value from memory		Data size in 2nd byte	not needed for 16/32-bit floats
ST	0.5	Store truncated to memory		Data size in 2nd byte	
STF		Store floating-point value to memory		Data size in 2nd byte	not needed for 16/32-bit floats
ADD	1	Unsigned binary add		Carry goes to residue register	Three operand instructions can access residue register
SUB	1	Unsigned binary subtract		Carry goes to residue register	Three operand instructions can access residue register
ADS		Signed binary add		Signed overflow goes to residue register	Three operand instructions can access residue register
SUBS		Signed binary subtract		Signed overflow goes to residue register	Three operand instructions can access residue register
AND	1	Bit by bit Boolean AND			
ANDN		Bit by bit Boolean AND between operand and 2nd complemented operand			a AND NOT b
OR	1	Bit by bit Boolean OR			
XOR	1	Bit by bit Boolean exclusive or			
MUL	1	Unsigned binary multiply		Upper half of product goes to residue register	Three operand instructions can access residue register
MULS		Signed binary multiply		Upper half of product goes to residue register	Three operand instructions can access residue register
DIV	1	Unsigned binary divide		Remainder goes to residue register	Three operand instructions can access residue register
IDIV		Signed binary divide		Remainder goes to residue register	Three operand instructions can access residue register
LSL	0.5	Logical shift left		Shifted out bits go to residue register	Three operand instructions can access residue register
ROTL	0.5	Rotate left			
LSR	0.5	Unsigned shift right		Shifted out bits go to residue register	Three operand instructions can access residue register
ASR	0.5	Signed shift right		Shifted out bits go to residue register	Three operand instructions can access residue register
EXTRCT	0.5	Extract bit field from source			Uses 12-bit immediate or LSBs of operand
EXTRCTS	0.5	Extract and sign extend bit field from source			Uses 12-bit immediate or LSBs of operand
FADD	1	Floating-point addition			
FSUB	1	Floating-point subtraction			
FMUL	1	Floating-point multiply			
FDIV	1	Floating-point divide			
FCMP	1	Floating-point compare		Destination gets a word of status bits	
CMP	1	Integer compare		Destination gets a word of status bits	
CALLR	1	Immediate is relative location of subroutine			11 bit displacement on 16-bit two byte instructions
CALLRF		Immediate is relative location of subroutine		three byte instruction	11 bit displacement on 16-bit two byte instructions
BR	1	Relative branch			11 bit displacement on 16-bit two byte instructions
BRZ	1	Relative branch if TOS zero			11 bit displacement on 16-bit two byte instructions
BRNZ	1	Relative branch if TOS not zero			11 bit displacement on 16-bit two byte instructions
IMM	1	Push immediate value			11-bit signed immediate
		Adjust framer/stack pointer			Use RTN
		Push/restore frame pointer			
CALLF		2nd byte is frame size and stack size			
MAX					see three operand inst
MIN					see three operand inst
IN	0.5	Input from IO register			Memory mapped IO would free up an op-code
OUT	0.5	Output to IO register			Memory mapped IO would free up an op-code
Unary ops	0.5	Operand field used for secondary op-code	Funct(TOS) -> TOS	trancendentals, sqrt, pop cnt, ldzct, …	also float <-> integer conversions & rounding using the residue register
Byte codes	2	stack and loop single byte op-codes			
Ternary ops	5.5				Heavy restriction on negate/complement bits
43	**31.5**				