

Instruction type	Example	Instruction meaning
Data transfers	Move data between byte, half-word, word, or double-word operands; * is data type	
	mov*	Move between two operands
	movzb*	Move a byte to a half word or word, extending it with zeros
	move*	Move the 32-bit address of an operand; data type is last
	push*	Push operand onto stack
Arithmetic/logical	Operations on integer or logical bytes, half words (16 bits), words (32 bits); * is data type	
	add*_	Add with 2 or 3 operands
	cmp*	Compare and set condition codes
	tst*	Compare to zero and set condition codes
	ash*	Arithmetic shift
	clr*	Clear
	cvtb*	Sign-extend byte to size of data type
Control	Conditional and unconditional branches	
	beql, bneq	Branch equal, branch not equal
	bleq, bgeq	Branch less than or equal, branch greater than or equal
	brb, brw	Unconditional branch with an 8-bit or 16-bit address
	jmp	Jump using any addressing mode to specify target
	aobleq	Add one to operand; branch if result \leq second operand
	case_	Jump based on case selector
Procedure	Call/return from procedure	
	calls	Call procedure with arguments on stack (see "A Longer Example: sort" on page E-33)
	callg	Call procedure with FORTRAN-style parameter list
	jsb	Jump to subroutine, saving return address (like MIPS jal)
	ret	Return from procedure call
Floating point	Floating-point operations on D, F, G, and H formats	
	addd_	Add double-precision D-format floating numbers
	subd_	Subtract double-precision D-format floating numbers
	mul f_	Multiply single-precision F-format floating point
	polyf	Evaluate a polynomial using table of coefficients in F format
Other	Special operations	
	crc	Calculate cyclic redundancy check
	insque	Insert a queue entry into a queue

FIGURE E.52 Classes of VAX instructions with examples. The asterisk stands for multiple data types: b, w, l, d, f, g, h, and q. The underline, as in addd_, means there are 2-operand (addd2) and 3-operand (addd3) forms of this instruction.