KE8-E EXTENDED ARITHMETIC ELEMENT

The KE8-E Extended Arithmetic Element (EAE) for the PDP-8/E enables the central processor to perform arithmetic operations at high speeds by incorporating the EAE components with the existing central processor logic circuitry so the two systems operate asynchronously. Most users employ the EAE in conjunction with the 23-bit Floating-Point Package, described in Chapter 2, to provide fast, convenient floating-point arithmetic, mathematical and trigonometric function evaluation, and formatted floating-point I/O. The option consists of two QUAD modules containing circuits that perform parallel arithmetic operations on positive binary numbers. It includes the registers and control logic circuits described in the following paragraphs.

Step Counter. The 5-bit step counter register is used to record the number of shifts performed during a logical or arithmetic shift operation and to stop the operation once the correct number of shifts has been executed. When an ASR, LSR, SCL, or SHL instruction is executed, the step counter is loaded with the complement of the step count contained in bits 7-11 of the memory location following the instruction. Bits 7-11 of the AC are loaded directly into the step counter during execution of an ACS instruction. The step counter is cleared for MUY, DVI and NMI instructions. The step counter is incremented as each shift is performed, and step counter overflow terminates the shift operation.

EAE Instruction Register. The EAE IR is a 12-bit register that is loaded during the FETCH cycle of EAE instruction execution. Bits 6 and 8-10 of the EAE IR are of particular interest, since these bits identify the particular EAE operation to be executed. (Bits 4, 5 and 7 are used by the group 3 operate instructions described in Chapter 3.)

EAE Timing and Control Logic. The EAE control logic is contained on modules which plus into the PDP-8/E OMNIBUS. These circuits are used in conjunction with the accumulator, link, multiplier quotient and memory buffer registers of the basic PDP-8/E to perform asynchronous arithmetic operations. The EAE control logic adds a larger class of arithmetic instructions to the group 3 operate instruction list.

EAE Mode Flip-Flop. The state of the EAE mode flip-flop determines which of two subsets of EAE instructions is currently implemented. The mode flip-flop is set to mode A when power is applied to the machine, when the CLEAR key on the programmer's console is operated, and when a CAF instruction is executed. It may be set to mode B or reset to mode A by certain EAE instructions.

Greater Than Flag. The greater than flag (GTF, not to be confused with GTF instruction) is a 1-bit register that is activated during execution of mode B EAE instructions. The GTF remains cleared during execution of all mode A instructions. When the GTF is activated, it receives the content of MQ11 during right shift operations. This facilitates subsequent round-off by indicating whether the content of the MQ should be rounded up (GTF set) or left alone (GTF cleared). The GTF is also set during execution of an SAM instruction, whenever the signed number in the MQ at the end of the operation is greater than or equal to the signed number that was in the AC at the beginning of the operation.

Programming the Extended Arithmetic Element

Extended Arithmetic Element instructions are an extension of the group 3 microinstructions. Like the other group 3 microinstructions introduced in Chapter 3, they have an OP-code of 7, while bits 3 and 11 are both set to contain binary ones. Mode A instructions are wholly compatible with PDP-8/I extended arithmetic element instructions, so that programs written for the PDP-8/I extended arithmetic element may run on the KE8-E Extended Arithmetic Element without modification. Mode B provides a greatly expanded set of instructions that is available for new programming on the PDP-8/E. Several EAE operations may be executed in either mode. The common features of these operations are described below.

Multiplication

During a multiplication operation, the content of the 12-bit MQ register is multiplied by a 12-bit multiplier (whose location depends upon the instruction mode). At the conclusion of the multiplication, the 12 most significant bits of the product are in the accumulator while the 12 least significant bits are in the MQ register. The multiplication is an unsigned integer multiply. That is, multiplier and multiplicand are treated as 12-bit, positive binary numbers with the binary point positioned after the least significant bit of each. The binary point of the product is positioned after the least significant bit of the MQ register. If the accumulator is non-zero at the start of the multiplication, its content is added to the low-order half of the product (contained in the MQ register). The link is always cleared.

Division

During a division operation, the content of the AC and MQ registers is treated as a 24-bit dividend with the 12 high-order bits in the AC. This number is divided by a 12-bit divisor (whose location depends upon the instruction mode) and the quotient and remainder are left in the MQ and AC registers, respectively. The division is an unsigned integer divide. The link is cleared if the first subtraction produces a negative result, indicating that divide overflow has not taken place. If the first subtraction produces a positive result, the link is set to indicate that divided overflow has occurred, and the division operation is terminated immediately. The content of the AC and MQ registers is modified by divide overflow, even though the operation is terminated prematurely. Thus, the divide instruction is ordinarily followed by a test of the link to check for overflow before further computation occurs.

Left Shift

During a left shift operation the link, AC and MQ are treated as one long register, with a high-order bit in the link and low-order bits in the MQ. The previous content of the link is lost during each shift, while AC0 is shifted into the link, MQ0 is shifted into AC11, and a zero is shifted into MQ11. The number of shifts to be executed is determined by a shift count contained in bits 7-11 of the location following the left shift instruction. Program execution resumes at the location following the shift count.

Logical Right Shift
During the logical right shift operation the link, AC and MQ are treated as one long register. MQ11 is either lost or shifted into the GTF, depending upon the mode of the instruction. AC11 is shifted into MQ0, while a zero is shifted into the link and into AC0. As in a left shift, the number of shifts to be executed is determined by a shift count contained in bits 7-11 of the location following the logical right shift instruction. Program execution resumes at the location following the shift count.

Arithmetic Right Shift
The arithmetic right shift operation is identical to the logical right shift except that AC0 is shifted into itself and into the link.

Normalization
Normalization is the process of converting a number with a known binary point into a fraction and an exponent. The step counter is initially cleared. The content of the link, AC and MQ are then shifted left, as described above, until AC0 and AC1 are different. The step counter is incremented once for each shift. (If AC2 through MQ11 are all zero, the number is already normalized and no shift occurs.) At the conclusion of a normalize operation, the step counter contains the binary number by which the AC and MQ were multiplied to accomplish normalization. Normalize instructions must not be microprogrammed with other instructions because the resulting octal codes are reserved to switch instruction modes.

The standard group 3 operate instructions introduced in Chapter 3 are implemented by logic circuitry in the PDP-8/E central processor. Table 3-8 lists these instructions, all of which are available even if an EAE is not installed. The additional EAE instructions described below may be considered as an extension of the group 3 operate instruction set. The extended instructions are implemented by circuitry contained in the KE8-E Extended Arithmetic Element.

KE8-E mode changing instructions are available in either mode of operation. Table 5-1 lists the KE8-E mode changing instructions, their mnemonics, and the operations they perform.

Table 5-1 EAE Mode Changing Instructions

MNEMONIC	OCTAL	OPERATION
SWAB	**7431**	**Switch from A to B.** If the mode flip-flop was set to A, it is set to B. If the mode flip-flop was already set to B, it remains in mode B. In either case, an MQL instruction is also executed.
SWBA	**7447**	**Switch from B to A.** If the mode flip-flop was set to B, it is set to A. If the mode flip-flop was already set to A, no operation occurs.

The following instruction sequence is used to test the EAE mode flip-flop and determine which mode is currently implemented:

MNEMONIC	OCTAL
CAM	7621
DPSZ	7451

A skip will occur if the EAE is in mode B. If the EAE is in mode A, the skip will not occur and the SC will be loaded into the AC and normalized (a meaningless operation that modifies the content of the AC).

Mode A Instructions

Figure 5-1 shows the format of a KE8-E mode A instruction. The OP-code must be 7, while bits 3 and 11 are both 1. Bits 4, 5 and 7 are used by the group 3 operate microinstructions introduced in Chapter 3. Bit 6 is set to indicate an SCA instruction. Bits 8-10 are set to indicate one of the mode A instructions listed in Figure 5-1. These instructions may be microprogrammed with SCA and the group 3 microinstructions to form non-conflicting combined operations, except where indicated in Figure 5-1. The microprogrammed combination of two (or more) extended arithmetic element instructions is the bitwise logical OR or the octal codes for the individual instructions.

Most of the mode A EAE instructions require an operand, which is assumed to occupy the next word in memory, following the instruction. After execution of an EAE instruction that requires an operand, program execution resumes at the memory location following the operand. The greater than flag (GTF), explained in more detail in the next section, is always zero for mode A instructions. Table 5-2 lists the mode A instructions, their mnemonics and the operations they perform.

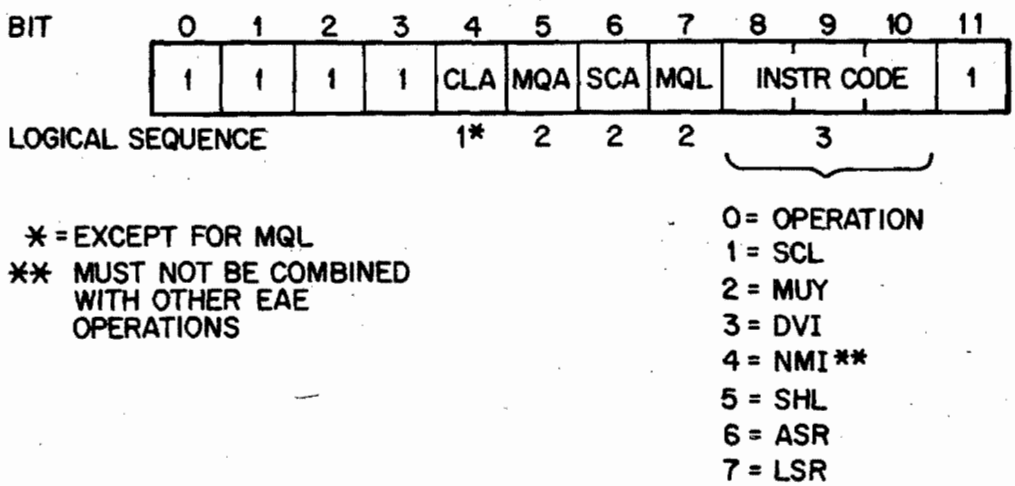

Figure 5-1 EAE Mode "A" Bit Assignments

Table 5-2 KE8-E Mode A Instructions

MNEMONIC	OCTAL	OPERATION
SCA	**7441**	**Step Counter OR with AC.** The content of the step counter is combined with the content of the low-order 5 bits of the AC (AC7-11) by a bitwise logical OR operation, and the result is loaded into AC7-11. AC0-6 remain unchanged.
SCA CLA	**7641**	**Step Counter to AC.** The content of the step counter is loaded directly into AC7-11. AC0-6 are cleared. This instruction is a microprogrammed combination of SCA and CLA.
SCL	**7403**	**Step Counter Load from Memory.** The next work in memory is treated as an operand. The one's complement of the low-order 5 bits of this operand (bits 7-11) is loaded into the step counter, and program execution resumes at the location following the operand. The SCL instruction is most commonly used during interrupt servicing, to restore the content of the step counter.
MUY	**7405**	**Multiply.** The next word in memory is taken as a multiplier. Multiplication occurs as described above, and program execution resumes at the location following the multiplier.
DVI	**7407**	**Divide.** The next word in memory is taken as a divisor. Division occurs as described above, and program execution resumes at the location following the divisor. If divide overflow occurs, the link is set. If the division was legal, the link is cleared.
NMI	**7411**	**Normalize.** The content of the AC and MQ are normalized as described above. This instruction must not be microprogrammed with any other instruction.
SHL	**7413**	**Shift left.** The content of the AC and MQ is shifted left as described above. The number of shifts performed is equal to one more than the content of the 5 low-order bits (bits 7-11) of the next location in memory. Program execution resumes at the location following the shift count.
ASR	**7415**	**Arithmetic Shift Right.** The content of the link, AC and MQ are shifted right as described above. The number of shifts performed is equal to 1 more than the content of the 5 low-order

Table 5-2 KE8-E Mode A Instruction (Cont.)

MNEMONIC	OCTAL	OPERATION
		bits (bits 7-11) of the next location in memory. The previous content of MQ11 is lost as each shift is executed. Program execution resumes at the location following the shift count.
LSR	**7417**	**Logical Shift Right.** The content of the link, AC and MQ are shifted right as described above. The number of shifts performed is equal to 1 more than the content of the 5 low-order bits (bits 7-11) of the next location in memory. The previous content of MQ11 is lost as each shift is executed. Program execution resumes at the location following the shift count.

Mode B Instructions

Mode B instructions differ from mode A instructions in the use of bit 6 of the instruction word, the location of operands, and in greatly increased double-precision arithmetic capability. Figure 5-2 shows the format of a mode B instruction. As with mode A instructions, mode B instructions may be microprogrammed to combine non-conflicting logical operations.

Some mode B instructions require a double precision operand, which is simply two consecutive memory locations that are assumed to contain a 24-bit number with the 12 most significant bits in the location having the lower memory address. A double precision operand is addressed by specifying the 12-bit address of the high-order half of the operand.

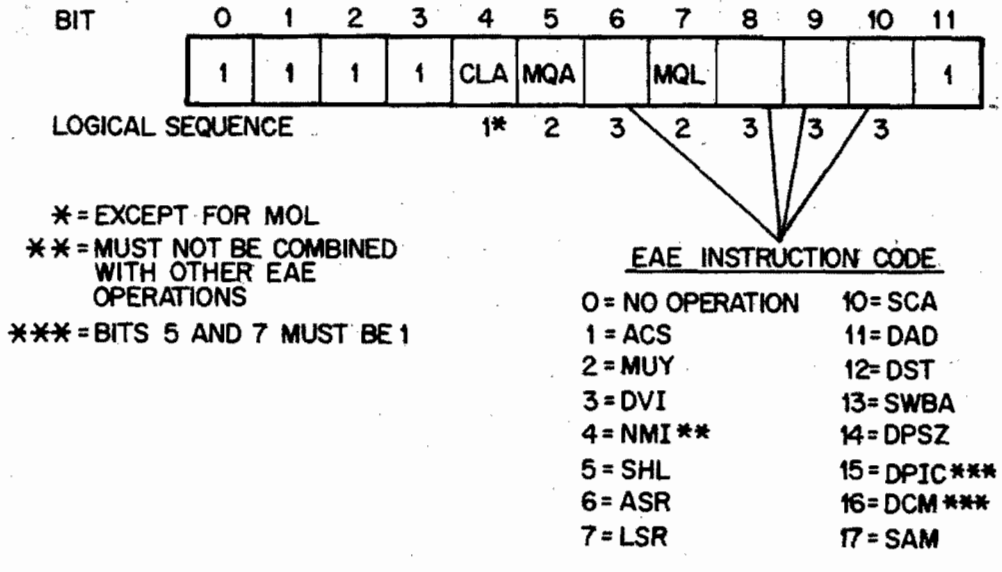

Figure 5-2 EAE Mode "B" Bit Assignments

The Greater Than Flag (GTF) is activated during execution of mode B instructions. The GTF may be manipulated by means of processor IOT instructions described in Chapter 4. It is conditionally loaded by the SAM instruction, and it receives the content of MQ11 during right shift operations. Table 5-3 lists the mode B instructions, their mnemonics, and the operations they perform.

Table 5-3 KE8-E Mode B Instructions

MNEMONIC	OCTAL	OPERATION
ASC	**7403**	**Accumulator to Step Count.** The low-order 5 bits of the AC (AC7-11) are loaded into the step counter, and the AC is then cleared.
MUY	**7405**	**Multiply.** The next word in memory is taken as the address of a multiplier. If extended memory is installed, the multiplier is obtained from the current data field. Multiplication occurs as described above, and program execution resumes at the location following the address of the multiplier.
DVI	**7407**	**Divide.** The next word in memory is taken as the address of a divisor. If extended memory is installed, the divisor is obtained from the current data field. Division occurs as described above, and program execution resumes at the location following the address of the divisor. If divide overflow occurs, the link is set. If divide overflow does not occur, the link is cleared.
NMI	**7411**	**Normalize.** The content of the AC and MQ is normalized as described above. This command must not be microprogrammed with any other instruction.
SHL	**7413**	**Shift Left.** The content of the AC and MQ is shifted left as described above. The number of shifts performed is equal to the content of the 5 low-order bits (bits 7-11) of the next location in memory. A shift count of zero is legal, and leaves the link, AC, and MQ registers unchanged. Program execution resumes at the location following the shift count.
ASR	**7415**	**Arithmetic Shift Right.** The link is loaded from AC0 and remains unaltered for the remainder of the operation. The content of the AC and MQ is then shifted right as described above. The number of shifts performed is equal to the content of the 5 low-order bits (bits 7-11) of the next location in memory. A shift count of zero is legal, and loads the link from AC0 but leaves the AC and MQ registers unchanged.

Table 5-3 KE8-E Mode B Instructions (Cont.)

MNEMONIC	OCTAL	OPERATION
		Bits shifted out of MQ11 are shifted into the GTF, to facilitate round-off operations. Program execution resumes at the location following the shift count.
LSR	**7417**	**Logical Shift Right.** The link is cleared and remains unaltered for the remainder of the operation. The content of the AC and MQ is shifted right as described above. The number of shifts performed is equal to the content of the 5 low-order bits (bits 7-11) of the next instruction in memory. A shift count of zero is legal, and clears the link without changing the AC or MQ registers. Bits shifted out of MQ11 are shifted into the GTF to facilitate round-off operations. Program execution resumes at the location following the shift count.
SCA	**7441**	**Step Counter OR with AC.** The content of the step counter is combined with the content of the low-order 5 bits of the AC (AC7-11) by a bitwise logical OR operation, and the result is loaded into AC7-11. AC0-6 remain unchanged.
SCA CLA	**7641**	**Step Counter to AC.** The content of the step counter is loaded into AC7-11. AC0-6 are cleared. This instruction is a microprogrammed combination of SCA and CLA.
SAM	**7457**	**Subtract AC from MQ.** The content of the AC is subtracted from the content of the MQ in two's complement arithmetic. The result is loaded into the AC. The MQ remains unchanged. If a borrow is propagated from the most significant bit, the link is set. Otherwise, the link is cleared. Hence, the link is set if and only if the original content of the AC was less than or equal to the content of the MQ. The GTF is helpful when comparing signed numbers. It is set if the signed number in the MQ is greater than or equal to the original signed number in the AC, and cleared otherwise.
DAD	**7443**	**Double Precision Add.** The double precision word addressed by the next memory location is added to the previous content of the AC and MQ registers. If extended memory is installed, the double-precision word is obtained from the current data field. If there is a carry from the most significant bit, the link is set. If there is

Table 5-3 KE8-E Mode B Instructions (Cont.)

MNEMONIC	OCTAL	OPERATION
		no carry, the link is cleared. Program execution resumes at the memory location following the operand address. This instruction may be microprogrammed with the CAM instruction to produce a double precision load (DLD) instruction.
DST	**7445**	**Double Precision Store,** The content of the MQ and AC is stored at the double precision location addressed by the next memory location. If extended memory is installed, the storage location will be in the current data field. The AC, MQ and link remain unchanged. Program execution resumes at the location following the operand address. This instruction may be microprogrammed with the CAM instruction to produce a Double Precision Deposit Zero (DDZ) instruction.
DPIC	**7573**	**Double Precision Increment.** The double precision constant "one" is added to the double precision number in the AC and MQ by two's complement arithmetic. The high-order carry (or lack thereof) is propagated into the link. This instruction requires that the MQL and MQA bits be set to function as defined.
DCM	**7575**	**Double Precision Complement.** The content of the AC and MQ, considered as a 24-bit number, is complemented and incremented. This has the effect of replacing the content of the AC and MQ with its two's complement. The high-order carry (or lack thereof) is propagated into the link. This instruction requires that the MQL and MQA bits be set in order to function as defined.
DPSZ	**7451**	**Double Precision Skip if Zero.** The double precision number contained in the AC and MQ is tested. If all bits are zero, the PC is incremented to skip the next sequential instruction. If any bit is 1, the next instruction is executed.

Table 5-4 lists the major differences between mode A and mode B instructions.

Table 5-4 Mode A and Mode B Instruction Differences

INSTRUCTION	MODE A	MODE B
MUY	The next location holds the multiplier.	The next location holds the address of the multiplier.
DVI	The next location holds the divisor.	The next location holds the address of the divisor.
SHL, LSR, ASR	The next location holds one less than the number of shifts. On Right Shifts, MQ11 is lost.	The next location holds the number of shifts. (A shift of zero places is legal). On Right Shifts, MQ11 is shifted into the GT flag.

Figure 5-4 summarizes cycle times and indicates the longest practical machine cycle. Note that the longest cycle time plus 0.3 μs. is the maximum time to enter a DMA cycle, provided the Break Device synchronizes at Int. Strobe time as recommended in Chapter 9. It is possible, by a small amount of programming, to reduce the longest cycle to 6.2 μs. This programming consists of pretesting the AC on a normalize, and limiting long shifts to 15 places. Note, for example, that

```
MQL          /AC MQ, O AC
LSR          /Mode B Shift, 6 places
6
```

is equivalent to an 18-bit logical right shift and has a longest cycle of 3.5 μs., rather than 7.1 μs. Also, the total execution time for a straight 18-bit shift is 8.3 μs., as opposed to 5.9 μs. for the above sequence.

	MODE A			MODE B			
	MEM CYCLES	INSTR TIME	LONGEST CYCLE	MEM CYCLES	INSTR TIME	LONGEST CYCLE	NOTES
SWAB	1	1.2μs	1.2μs	1	1.2	1.2	
SWBA	1	1.2	1.2	1	1.2	1.2	
SCL	2	2.6	1.4	Not Available			
ACS	Not Available			1	1.2	1.2	
MUY	2	7.4	6.2	3	8.6	6.2	
DVI	2	7.4	6.2	3	8.6	6.2	No overflow
NMI	1	1.5+.3N	8.1	1	1.5+.3N	8.1	
SHL	2	2.6+.3N	8.9*	2	2.9+.3N	9.2**	25-place shift
ASR	2	2.6+.3N	8.9*	2	2.9+.3N	9.2**	25-place shift
LSR	2	2.6+.3N	8.9*	2	2.9+.3N	9.2**	25-place shift
SCA	1	1.2	1.2	1	1.2	1.2	
DAD	Not Available			4	5.2	1.4	
DST	Not Available			4	5.2	1.4	
DPSZ	Not Available			1	1.2	1.2	
DPIC	Not Available			1	1.6	1.6	
DCM	Not Available			1	1.6	1.6	
SAM	Not Available			1	1.2	1.2	

*Computed from 1.4+.3N
**Computed from 1.7+.3N