PDP-1 COMPUTER
ELECTRICAL ENGINEERING DEPARTMENT
M. I. T.
CAMBRIDGE 39, MASSACHUSETTS

May 18, 1965

Modification Bulletin No. 1-1

The Electrical Engineering Department and Research Laboratory of Electronics PDP-1 is equipped with several additional instructions.

GENERAL INSTRUCTIONS

1. adm Y
   op code 36

   Add to memory. The new C(AC) and the new C(Y) are the sum of the original C(Y) and the original C(AC). The addition is performed with 1's complement arithmetic. If the sum of two like-signed numbers yields a result of the opposite sign, the overflow flip-flop will be set. A result of minus zero is changed to plus zero.

2. jdp Y
   op code 14

   Jump and Deposit Program Counter The contents of the Program Counter (holding the address of the instruction following the jdp) are deposited in the memory register Y. The original contents of the AC remain in the AC unchanged. The state of the overflow flip-flop is saved as described under jsp. The next instruction executed is taken from Memory Register Y+1.

SKIP GROUP - op code 64

	1. sni address 4000	skip on non-zero in-out. This is an additional skip class instruction, which may be or-ed with all other skip class instructions. Sni skips whenever the in-out register contains anything other than +0.
*	2. szm	skip on zero or minus accumulator. This instruction is not a new hardware feature; it is just defined in the initial symbol table to represent the or-ing of two instructions, sza and sma.
*	3. spq	skip on positive quantity. This instruction is not a new hardware feature; it is just defined in the initial symbol table to represent the or-ing of two instructions, sza and sma i.
*	4. clo	clear the overflow flip-flow. This flip-flop is set only by an addition or subtraction that exceeds the capacity of the AC. The overflow flip-flop is not cleared by any arithmetic operations. This instruction is not a hardware feature; it is just defined in the initial symbol table to represent the following or-ing: szoVsmaVspaVi

* Extended Instructions

OPERATE GROUP - op code 76

	1. lai address 40	load accumulator from in-out. This operate class instruction copies the contents of the in-out register into the AC. It happens after all normal operate class options (see timing diagram below). If the computer is stopped at the end of this instruction the MBR will contain zero, and the old contents of the AC will be shown.
	2. lia address 20	load in-out from accumulator. This operate class instruction copies the contents of the AC into the IO register. It happens after all normal operate class options, and the old contents of the IO will be shown. (see timing diagram below).
*	3. swp address 60	swap accumulator with in-out. This is the combination of the above two instructions. It happens after all normal operate options. If the computer is stopped of the end of this instruction, the MBR will contain zero, and the swap will not yet have occurred.
*	4. clc	clear and complement the accumulator. This instruction is not a new hardware feature; it is just defined in the initial symbol table to represent the or-ing of two instructions, cla and cma.

Order of operate class instructions:

EFFECT	0→AC 0→IO	TW→AC,stf PC→AC,clf	$\overline{AC}$→AC 0→RUN	0→MB (lai)	IO→MB(lai) 0→IO(lia)	AC→MB(lia) MB→AC(lai)	MB→IO (lia)
TIME	7	8	9	10	0	1	2
	Time pulses of this instruction				TPs of next instruction		

* Extended instructions

DRUM INSTRUCTIONS

1. dia(720060) — drum initial address. Causes the $C(IO)_{1-5}$ to be sent to the drum write field buffer. These bits specify which field of the drum will be written on during the next dcc instruction; or if $C(IO)_{1-5}=0$, that no write operation is to occur. The $C(IO)_{6-17}$ are sent to the drum initial address register to specify the first drum address to be transferred.

2. dba(720061) — drum break on address. Causes the $C(IO)_{6-17}$ to be sent to the drum initial address register. When the current drum address becomes equal to the contents of the initial address register, a sequence break request is indicated. Bit 5 of the status word is set by the break, and is cleared by the next dcc instruction.

3. dcc(720062) — drum count and commence. Causes the $C(IO)_{1-5}$ to be sent to the drum read field buffer. These bits specify which field will be read; or, if $C(IO)_{1-5}=0$, that no read operation is to occur. The $C(AC)_{6-17}$ specify the first core memory address of the data to be transferred. The $C(IO)_{6-17}$ specify the number of words to be transferred. If the $C(IO)_{6-17}=0$, 4096 words are transferred. While the dcc instruction is being executed, the computer stops and the drum system takes full control of the core memory. Successive words are transferred from sequential locations until the operation is complete. If no errors occurred during the drum operation, the instruction following the dcc will be skipped. The C(AC) and C(IO) are lost during this operation. If both the read field and write field are non-zero (both reading and writing operations are specified) the contents of memory are written on the write field; then the read field data are read into memory. The read field must not equal the write field.

4. dra(720063) — drum read address. Cause the current drum address to be read into the IO_{6-17}. The parity error flag is read into IO_0; the write error flag is read into IO_1; and the timing error flag is read into IO_2. Two cycles elapse before this information is placed in the IO.