Datatron 205 Fast Drum Zero Program--Decoded

Electrodata/Burroughs Datatron 205 program found pasted onto one of the notes pages at the end of Don Knuth's copy of the Burroughs 205 Central Computer Handbook, March 1957, revised February 1958.

Presumably this program is loaded with a CDR instruction similar to 0-0000-44-4000, although it turns out that any address could be used in the last four digits.

Note that this program assumes the presence of a Cardatron unit, and that the format digit (6) will be detected in one of the first three columns on the card. Format 6 is a straight numeric load of seven 11-digit words plus one 3-digit word, proceeding BACKWARDS on the card -- i.e., the first word comes from columns 70-80, the second from 59-69, the third from 48-58, etc., with the eighth word coming from columns 1-3. An additional six words of zeroes will be stored in memory following the eight data words, unless they are inhibited by an in-line command as discussed below. Format 6 ignores zone punches on the card, so the "B" and "D" in the raw data below are read as "2" and "4", respectively. The first card in the deck has format digit 7, which causes it to be ignored by the Cardatron unit.

2014-11-08 Paul Kimpel, original version.

```
777 FAST DRUM ZERO. USES 4000 LOOP. ZEROES 6000-6019 AND 0000-3999 IN 1.1 SEC.
666608104D4024000002040260000064403500000224017100000260100000072402740810444016
6666081044403010000B60100100002600601000026002000000724035000001240350000075403 4
6666000020401600000003900000000001000081044100000000224024100002601801000026014 0
```

True Addr	Instr Addr	Instruction	Label	Mnem	Notes
----	----	4 0810 44 4016		CDR 4016	Read card from unit 1 starting at address 4016, suppressing B-register modification. Since the sign digit is 4, however, this instruction is not stored in memory, but executed immediately as it is being read from the card. It effectively redirects the remaining words on the card to the new address, which is why the address used to load the program does not matter -- the program overrides that during the load process.
4016	4016	0 0000 72 4027	START	SB 4027	Set B to the low-order four digits of the word at 4027, which is 0020.
4017	4017	1 0000 02 6010	CLR6	STC 6010+B	Store contents of the A register at 6010, modified by the contents of the B register, then clear A. This and the next instruction form a loop to clear the 6000 loop to zero. The original contents of A do not matter, as the loop will be executed with values of B from 20 down to 0, or 21 times. Since this stores into the 6000 loop, addresses are taken modulo 20, thus the first word stored, at 6030 (=6010) will be overwritten again during the last iteration of the loop, when B=0 and A is known to be zero.
4018	4018	0 0000 22 4017		DB CLR6	If B>=0, decrement B and branch to 4017; otherwise decrement B (leaving it at 9999) and do not branch.
4019	4019	0 0000 64 4035		CAD 4035	Clear A and add the contents of 4035 to A. This word initially contains 3900, and is used to maintain the base address for the current iteration of BF6 ops below.
4000	4020	0 0000 20 4026		CU INIT	Branch unconditionally to 4026 to start the memory clearing process. Note that since addresses in the 4000 loop are interpreted modulo 20, this instruction is actually stored at address 4000.
----	----	6 0810 44 4024		CDR 4024	Read the next card of the program to address 4024. Since the sign digit is 6, this is another instruction that is executed immediately during load and is not stored in memory. It also stops data transfer from the Cardatron for the prior card, so the eighth word of data from columns 1-3 and the six words of zeroes that would normally be stored in memory are inhibited. Note that address 4024 is actually address 4004.
4004	4024	0 0000 75 4034	NEXTL	SU 4034	Subtract the contents of 4034 (a constant 100) from A.

4005	4025	0 0000 12 4035		ST 4035	Store A at 4035, overwriting the former base address for the BF6 instructions.
4006	4026	0 0000 72 4035	INIT	SB 4035	Load the base address to B. Initially this is 3900, but will be decremented by 100 each time through the "NEXTL" loop.
4007	4027	1 0000 26 0020		BF6 0020+B	Block transfer 20 words from the 6000 loop to address 0020, modified by the B register. Since the 6000 loop was cleared to zeros above, this effectively zeroes 20 words with one instruction.
4008	4028	1 0000 26 0060		BF6 0060+B	Block transfer 20 words from the 6000 loop to address 0060+B.
4009	4029	1 0000 26 0100		BF6 0100+B	Block transfer 20 words from the 6000 loop to address 0100+B.
----	----	6 0810 44 4030		CDR 4030	Read the next card of the program to address 4030. This is another instruction that is executed immediately during load and not stored in memory, terminating the data transfer from the current card. Note that address 4030 is actually 4010.
4010	4030	1 0000 26 0140		BF6 0140+B	Block transfer 20 words from the 6000 loop to address 0140+B.
4011	4031	1 0000 26 0180		BF6 0180+B	Block transfer 20 words from the 6000 loop to address 0180+B.
4012	4032	0 0000 22 4024		DB NEXTL	If B>=0, decrement B and branch to 4024; otherwise decrement B to 9999 and do not branch.
4013	4033	0 0810 44 1000		CDR 1000	Read the next card to address 1000. Note that this read has a sign digit of 0, so it is stored in memory and executed as part of the program after the "NEXTL" loop terminates. Presumably this would load the first card of a program deck that follows this program, and that program deck could modify the load address with a sign-4 or sign-6 instruction as this program did in the beginning.
4014	4034	0 0000 00 0100		--	Constant 100 used to decrement the base address
4015	4035	0 0000 00 3900		--	Base address for the BF6 instructions. This is initially 3900. It is decremented by 100 each time through the "NEXTL" loop and loaded into the B register to offset the addresses in the BF6 instructions.
	----	6 0000 20 4016		CU START	Branch to 4016 to start executing the program just loaded. This is another sign-6 word that is executed immediately during load and not stored to memory. It also stops the data transfer of the eighth word and six words of zeroes from the Cardatron, which would otherwise overwrite a good piece of the program just loaded. as the true addresses would wrap from 4019 to 4000.
					There is one other piece of 205 weirdness to note with this program. The B register is initially loaded with 3900 and then decremented by 100 each time through the "NEXTL" loop. Thus the sequence of addresses used by the BF6 instructions in the first iteration is 3920, 3960, 4000, 4040, 4080. The second iteration uses 3820, 3860, 3900, 3940, 3980. The third iteration uses 3720, 3760, 3800, 3840, 3880, and so on.
					It would appear that the address in the first iteration will overwrite the 4000-loop, which would of course destroy the program. Not so -- since it is not possible to block-transfer from loop to loop, only from loop to main memory or vice versa, addressing for the BFx and BTx instructions is taken strictly modulo-4000. Thus BF6 4000 is actually interpreted as BF6 0000.

					Finally, incrementing the addresses between the BF6 instructions by 40 instead of 20 speeds things up considerably. The BF6 ends on a modulo-20 address. If the increment between instructions was only 20 words, there is not time to fetch and execute the next BF6 instruction before its effective address passes under the drum heads -- we've already passed it. The next BF6 is at true address 4008, so those eight word-times give us time to fetch and set up that BF6 (four word-times, minimum, once address 4008 is reached) before the current block of 20 words passes under the heads. Thus the delay between BF6 instructions offset by 40 words is only 20 word-times, rather than at least 200 word-times, as it would be if increments of 20 words were used.