From the 709 C.E. Manual

4.00.00 INSTRUCTION EXECUTION

ALL INSTRUCTIONS follow a definite order. This sequence is set up as soon as an instruction is decoded and carries on until the execution is complete. All instructions start with an instruction cycle The type of cycle which follows is determined by the cycle control circuits, Systems section 8. All instruction cycles are similar and with few exceptions follow a basic pattern.

4.01.00 FUNDAMENTAL CONCEPTS

4.01.01 Basic I-E Cycles

The objectives of the instruction cycle are:

1. To obtain a new instruction from storage
2. To establish the execution control circuits
3. To prepare to read a factor from storage

Figure 28 shows the units involved in the instruction cycle. The cycle begins with the instruction counter, which furnishes the address of the next instruction to the storage. The instruction is read out of this address and placed in the storage register. The instruction register is reset and receives the operation code from the storage register, thus establishing the execution control through the operation decoders. The instruction address is then routed through the adders in preparation for either reading a word from storage or being set into the shift counter positions of the instruction register. Finally, the instruction counter is stepped to address the next instruction, and execution begins.

Figure 29 is a logical diagram of the instruction cycle operation. It illustrates the following table:

	Command	Instruction Cycle Sequence Timing	Conditions	Systems
I CYCLE				
1.	Go to I Time	CT10 to I6	End Op Tgr on	8.05.01
2.	IC to AS	CT10 to I6	End Op Tgr on	3.40
3.	AS to AR	A0 (D1)	Gated	3.42
4.	AR to SAR	I2 (D1)		1.05.03 (10.01.01)
5.	AR to IC	I3 (D1)		3.20.02
6.	Stor to SBR to SB			1.03.01 (10.11.01)
7.	SB (S-35) to SR	I7 (D1)		2.08.01 (1)
8a.	Reset IR (S-9) off	I8 (D1)		3.01
8b.	Reset IR (10-17) on	I8 (D3)		3.05 (1)
9a.	SR (S, 1, 2) to IR (S, 8, 9)	I9 (D1)	SR (1 or 2) Plus	3.01
9b.	SR (S, 3-11) to IR (S-9)	I9 (D1)	Not SR (1 or 2) Plus	3.01
10.	Operation decoded	I9.5 to I8		3.10
11.	SR (18-35) to ADD(P-17)	I9 to CT1		2.08.48 (1)
12.	ADD (P-17) to AS	I9 to CT1		3.40

Instruction Cycle Sequence (continued)

Command	Timing	Conditions	Systems
15. Advance Inst Ctr	I11 (D1)		3.20
16a. Go to ER Time	I12 (D1)*	IR (1,2) Minus or IR (1,2,3) Plus	8.05.01
16b. Go to E Time	I12 (D1)*	Not Go to ER Time	8.05.01
17. AS to IR (10-17)	ER0 (D1)	Pri Op 7,6	3.05 (2)
18. AS to AR	A0 (D1)	Gated	3.40
19. AR to SAR	E2 (D1)		1.05.03 (10.01.01)
20. Stor to SBR to SB			1.03.01 (10.11.01)
21. SB (S-35) to SR	E7 (D1)		2.08.01 (1)

* Effective time

4.01.02 Address Modification

Instruction addresses may be temporarily modified to permit instructions to operate on information from a variety of locations. This programming feature is the function of the index register. Modification occurs within the I9 to CT1 gate as the instruction address is being brought through the adders to the address register. If indexing is desired, for example, the 2's complement of the amount in an index register is brought to the adders at this time. As a result of this operation, the address switches receive what is known as the effective address, which is the difference between the address and the index register. Execution of the instruction then proceeds in the usual manner using the effective address.

Selection of the index register to be used is made by the tag bits of the instruction. If more than one index register is specified, the registers are combined by logical OR before being subtracted from the address.

The index register mixing circuit is shown on Systems 2.12.04. Any register output may be selected, inverted, and gated to the adders as illustrated on Systems 2.02.04 through 2.02.10. The index register to adder gate is developed on Systems 2.08.49.

4.01.03 Indirect Addressing

Another programming device which permits the changing of an instruction address is indirect addressing. Bits in positions 12 and 13 of an instruction are signals for indirect addressing This causes an E cycle, during which the word located at the original instruction address is read out of storage and its address portion is substituted for the instruction address. The new address then becomes the one upon which the instruction operates. Since only indexable instructions may have indirect addressing, the original address and the new address may be modified by indexing, if they have tag bits. All this may be confusing from the programming viewpoint but is is really a simple operation. The only change indirect addressing causes is the insertion of an E cycle. As with direct address instructions, indexing is automatic.

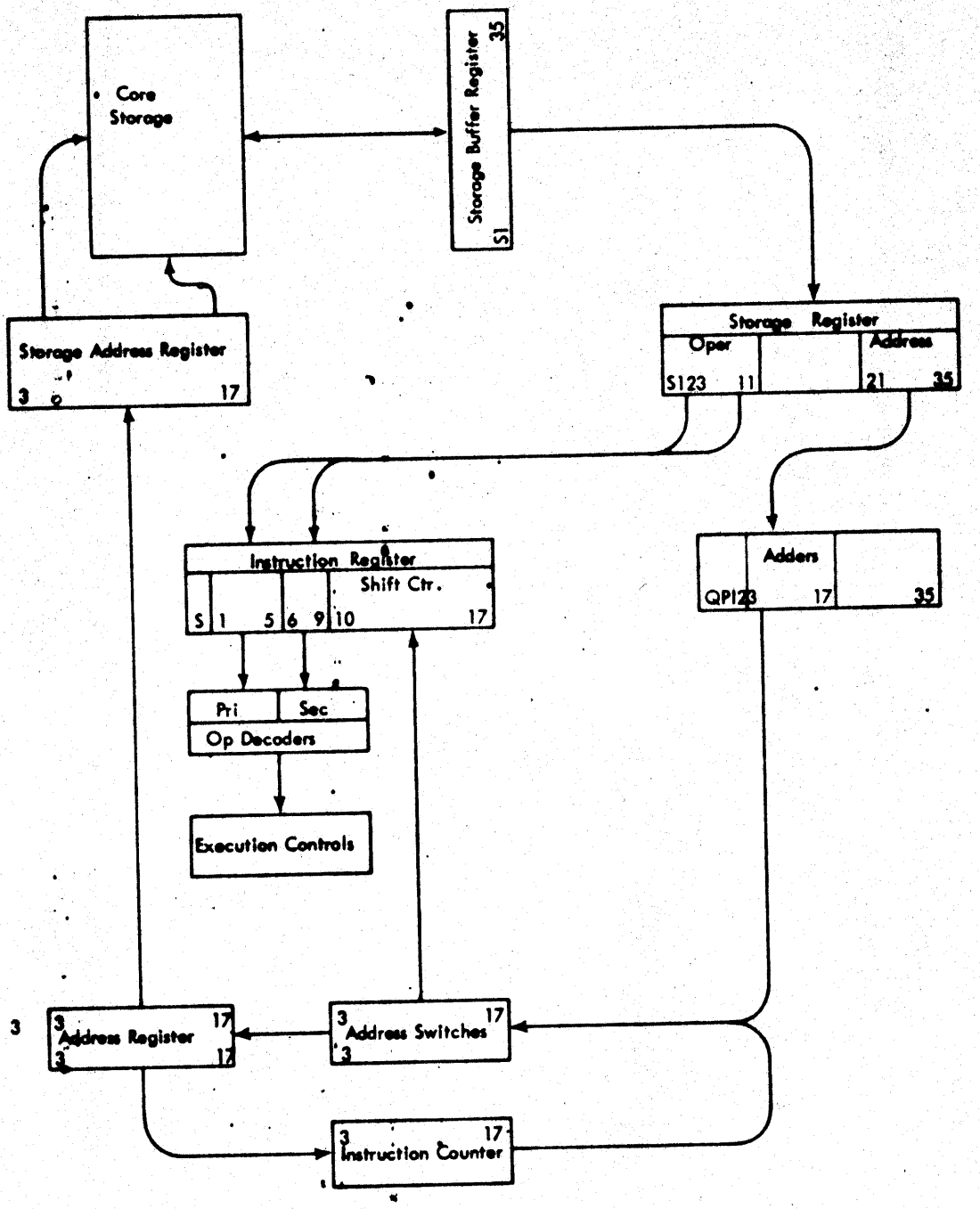

FIGURE 28. INSTRUCTION FLOW DIAGRAM

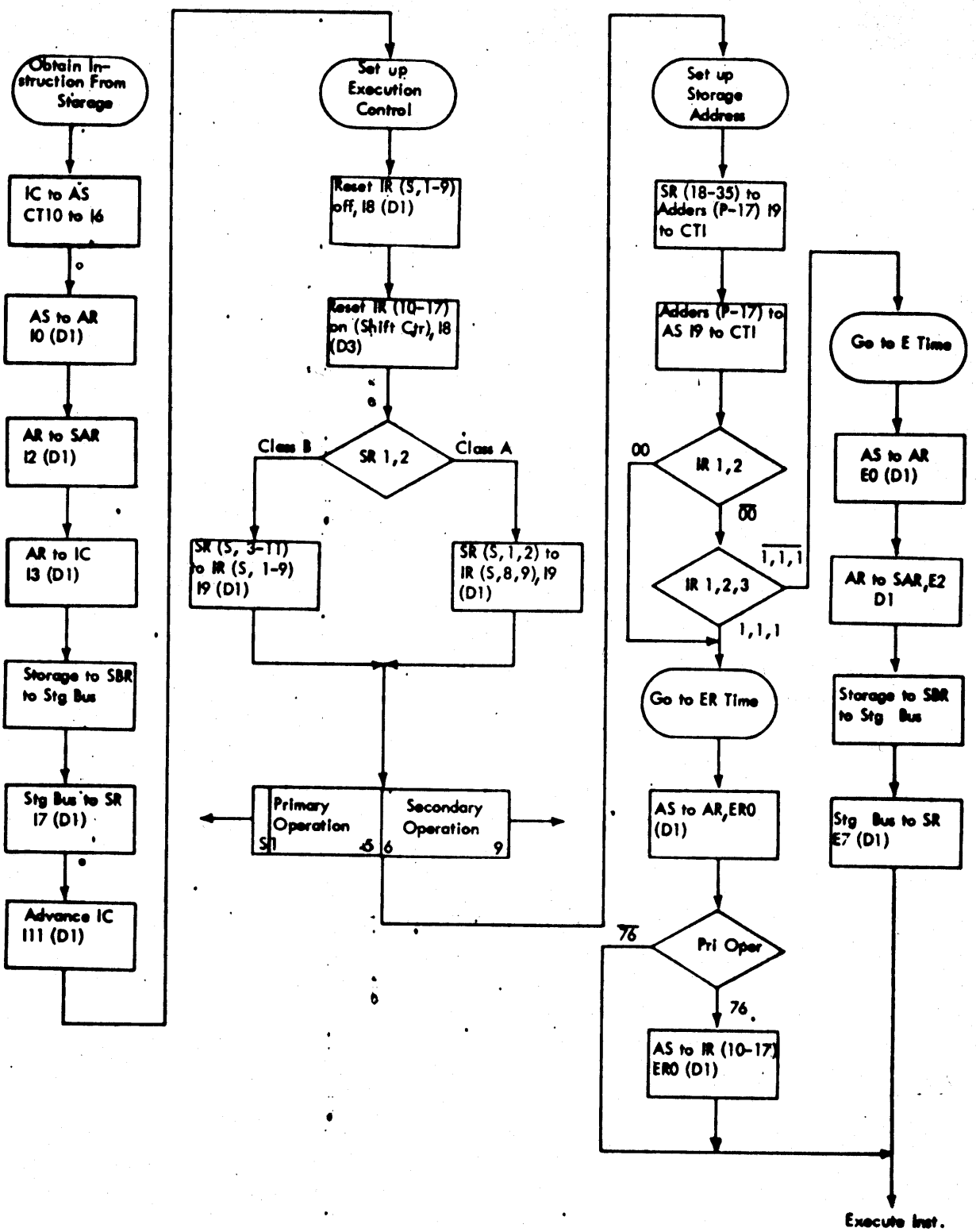

FIGURE 29. INSTRUCTION CYCLE