

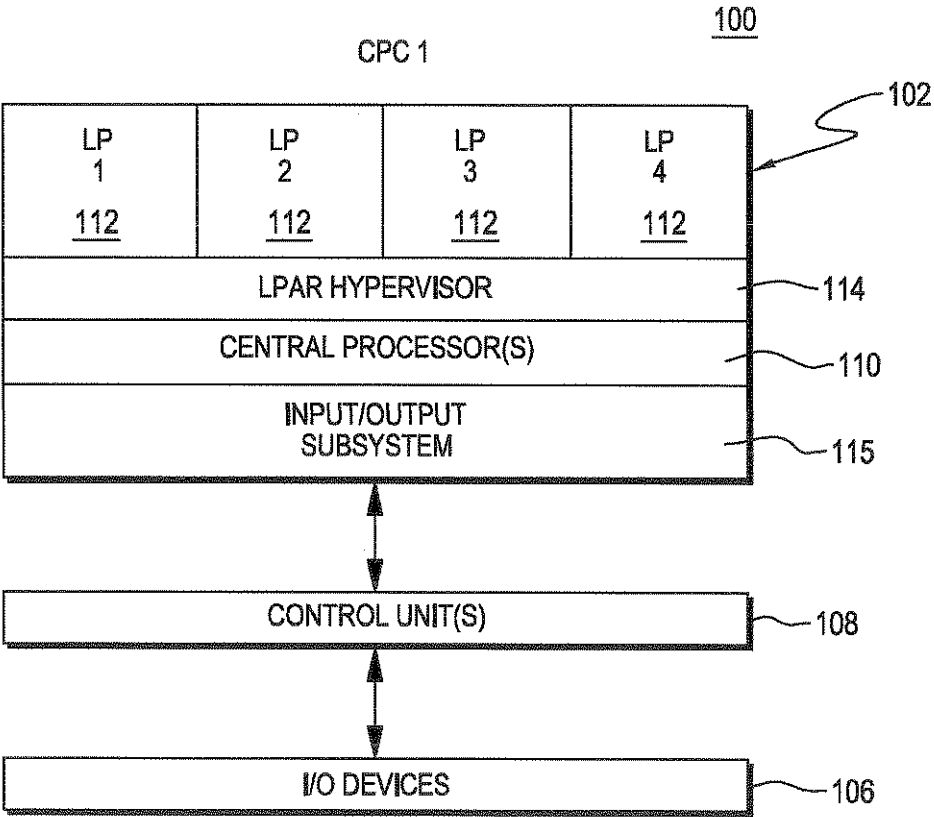


FIG. 1

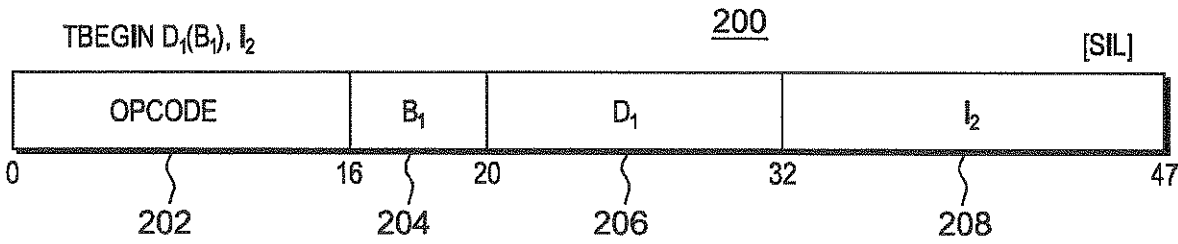


FIG. 2A

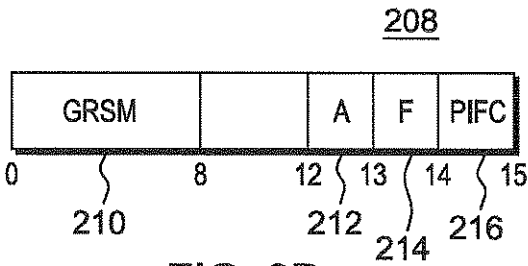


FIG. 2B

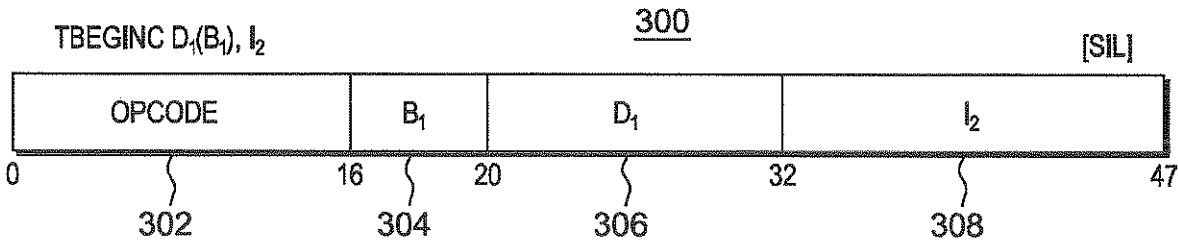


FIG. 3A

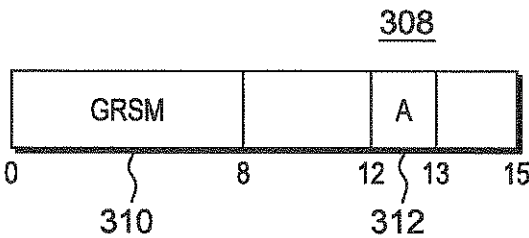


FIG. 3B

3/26

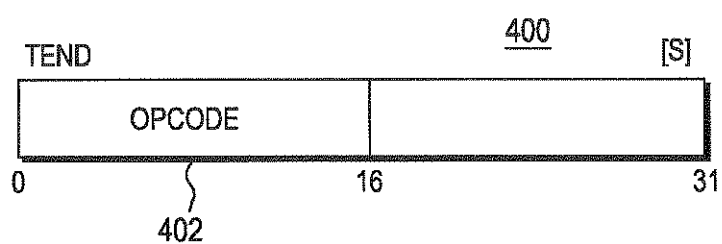


FIG. 4

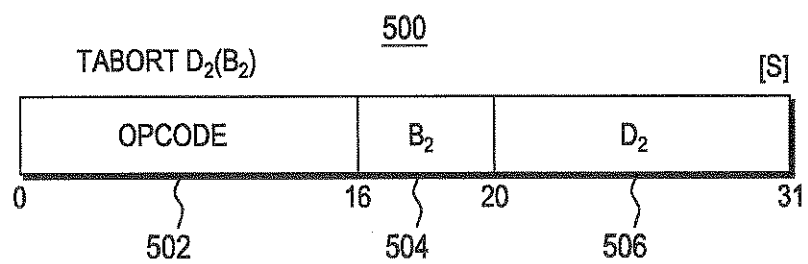


FIG. 5

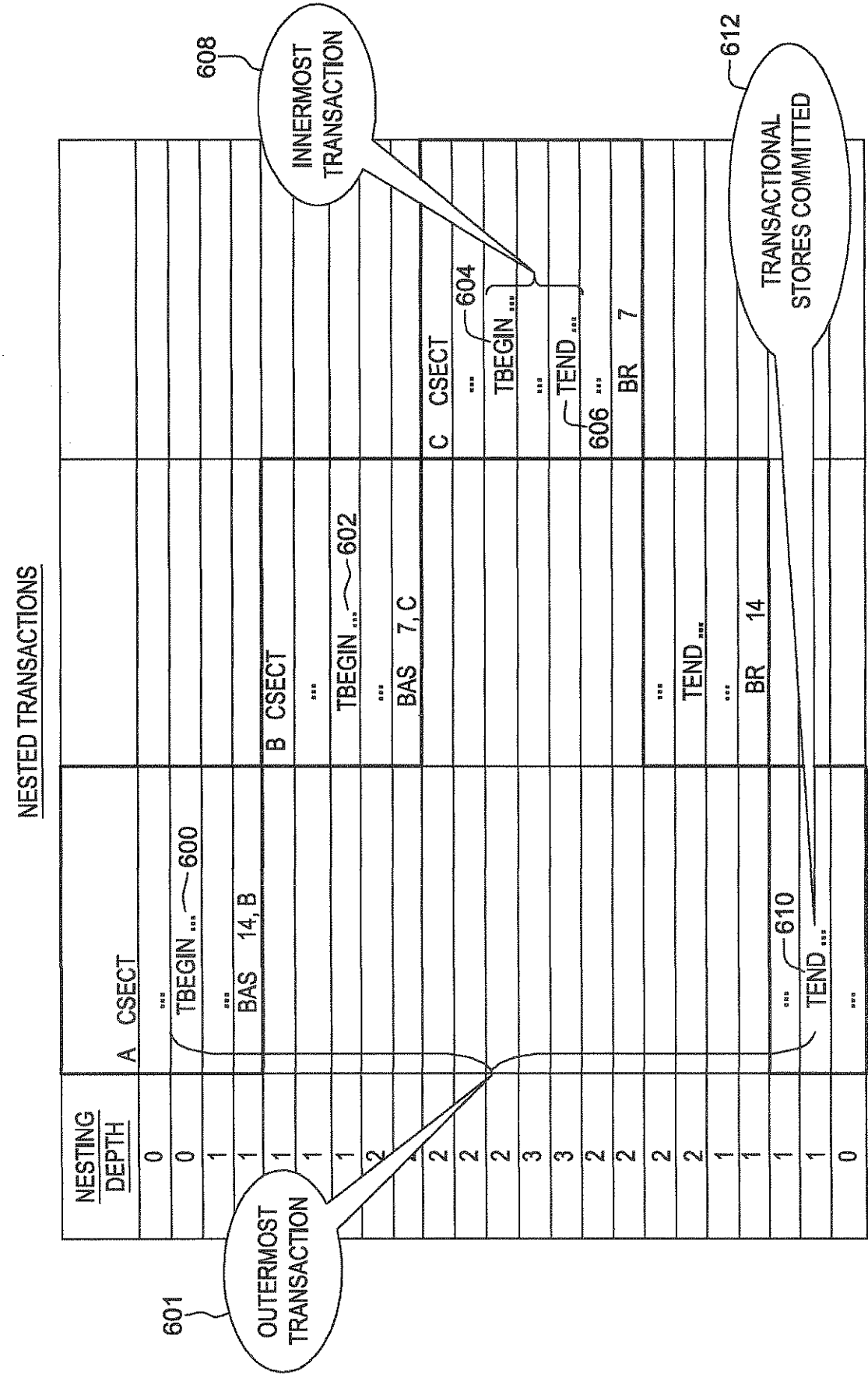


FIG. 6

5/26

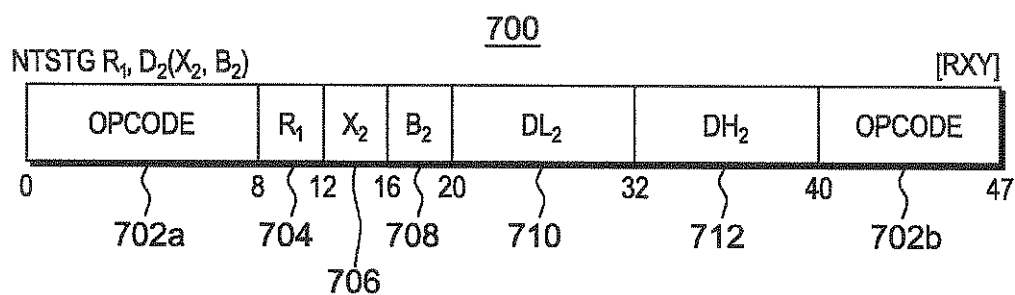


FIG. 7

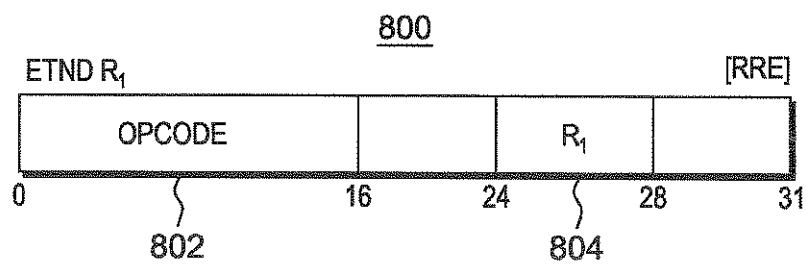


FIG. 8

6/26

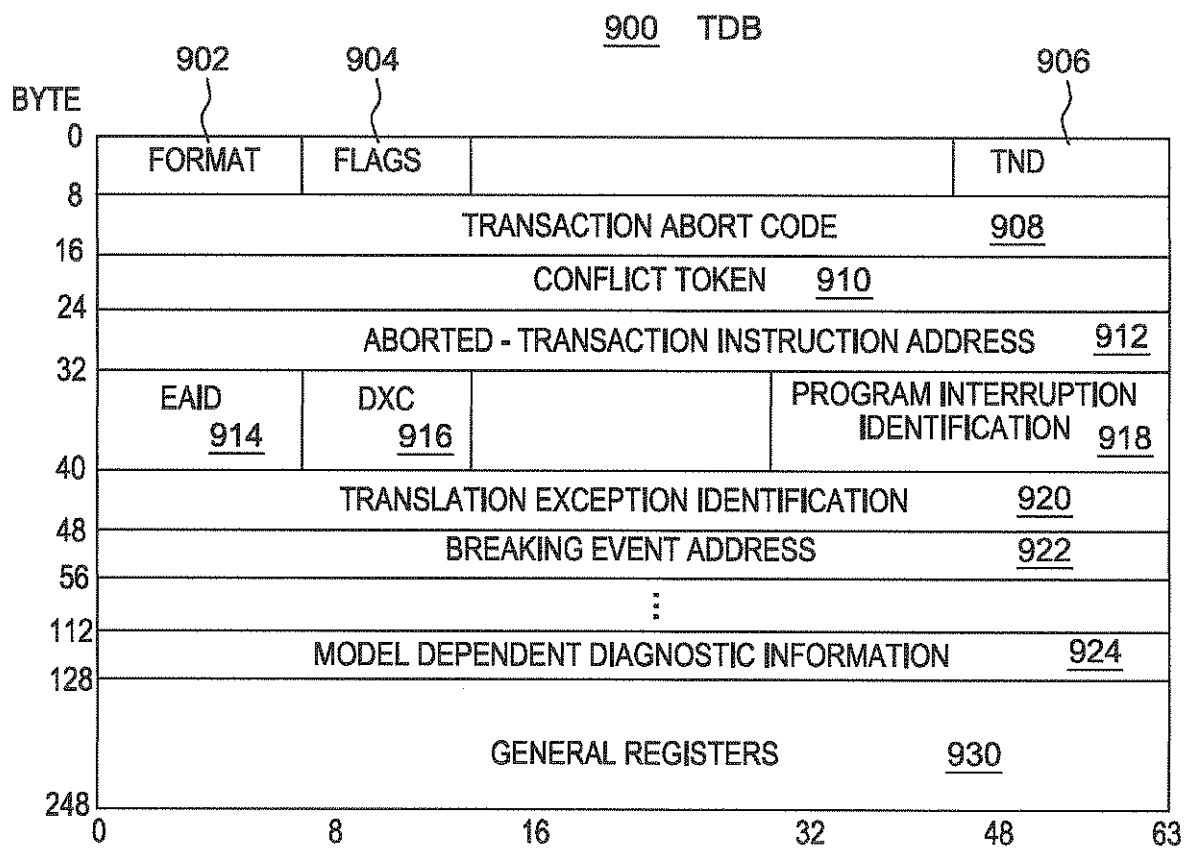


FIG. 9

7/26

CODE	REASON FOR ABORT	CC SET
2	EXTERNAL INTERRUPTION	2
4	PROGRAM INTERRUPTION (UNFILTERED)	2 OR 3 +
5	MACHINE - CHECK INTERRUPTION	2
6	I/O INTERRUPTION	2
7	FETCH OVERFLOW	2 OR 3
8	STORE OVERFLOW	2 OR 3
9	FETCH CONFLICT	2
10	STORE CONFLICT	2
11	RESTRICTED INSTRUCTION	3
12	PROGRAM EXCEPTION CONDITION (FILTERED)	3
13	NESTING DEPTH EXCEEDED	3
14	CACHE FETCH RELATED	2 OR 3
15	CACHE STORE RELATED	2 OR 3
16	CACHE OTHER	2 OR 3
255	MISCELLANEOUS CONDITION	2 OR 3
>255	TABORT INSTRUCTION	2 OR 3
‡	CANNOT BE DETERMINED; NO TDB STORED	1
EXPLANATION: + CONDITION CODE IS BASED ON INTERRUPTION CODE ‡ THIS SITUATION OCCURS WHEN A TRANSACTION ABORTS, BUT THE TDB HAS BECOME INACCESSIBLE SUBSEQUENT TO THE SUCCESSFUL EXECUTION OF THE OUTERMOST TBEGIN INSTRUCTION. NO TBEGIN SPECIFIED TDB IS STORED, AND THE CONDITION CODE IS SET TO 1.		

FIG. 10

8/26

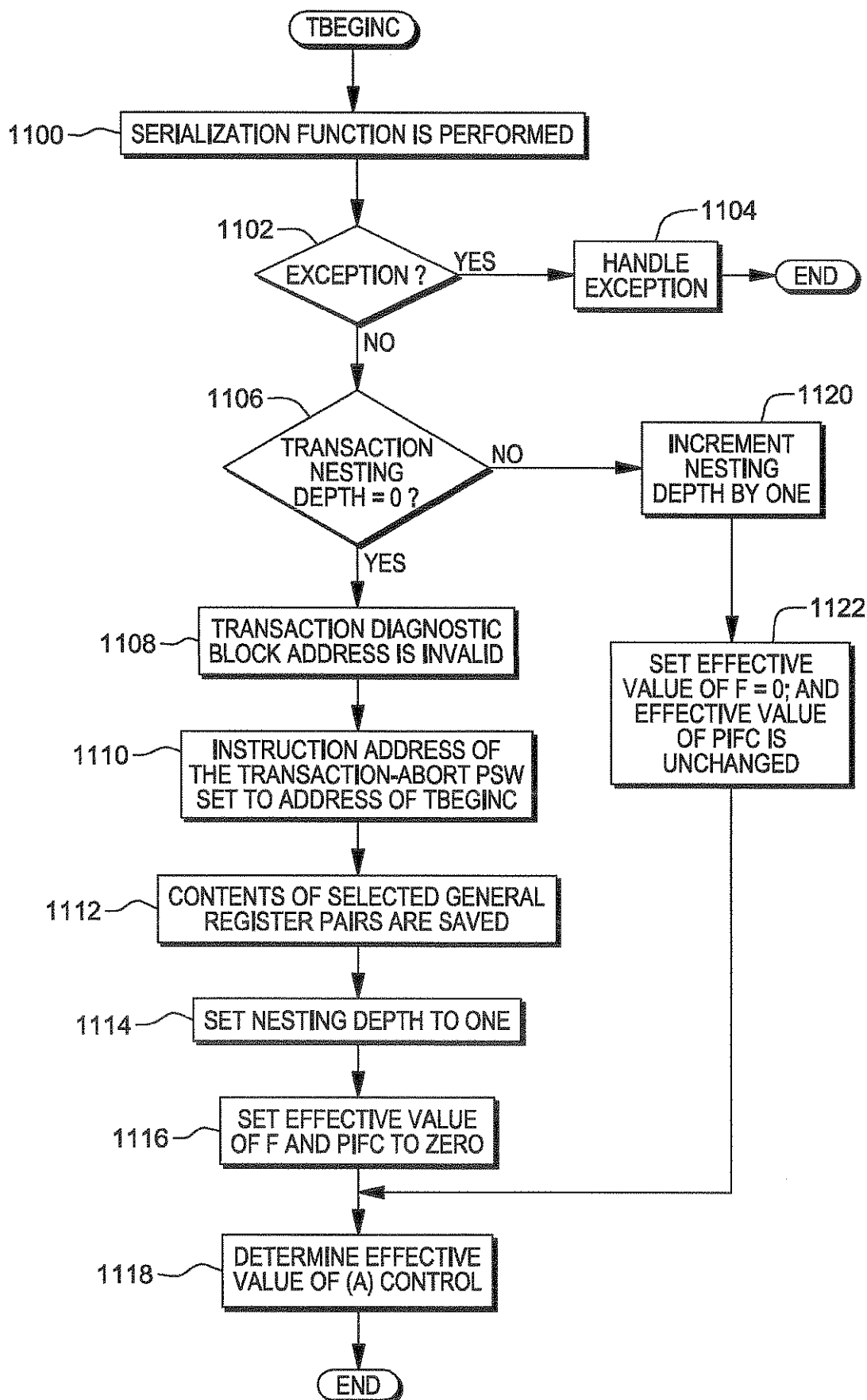


FIG. 11

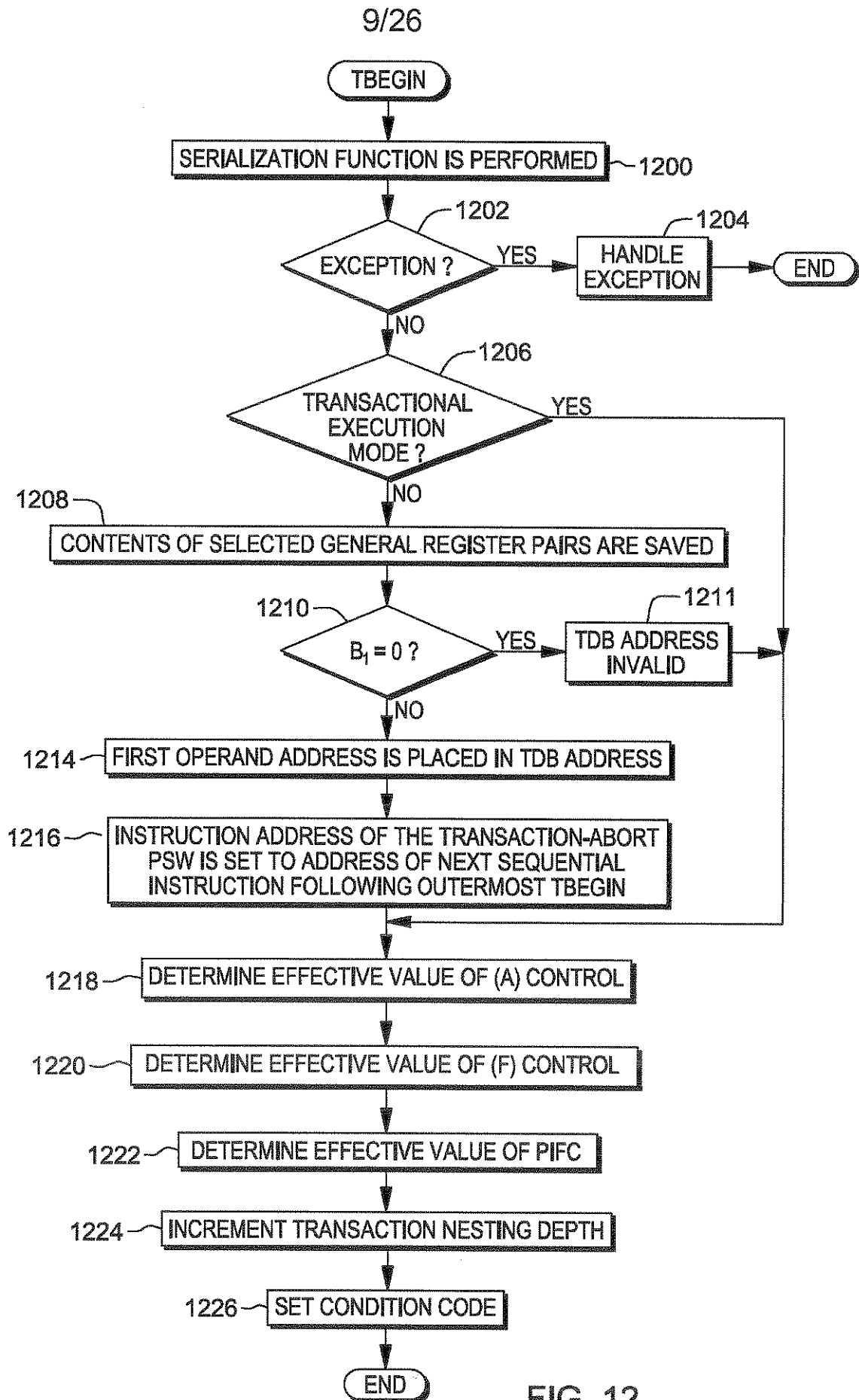


FIG. 12

10/26

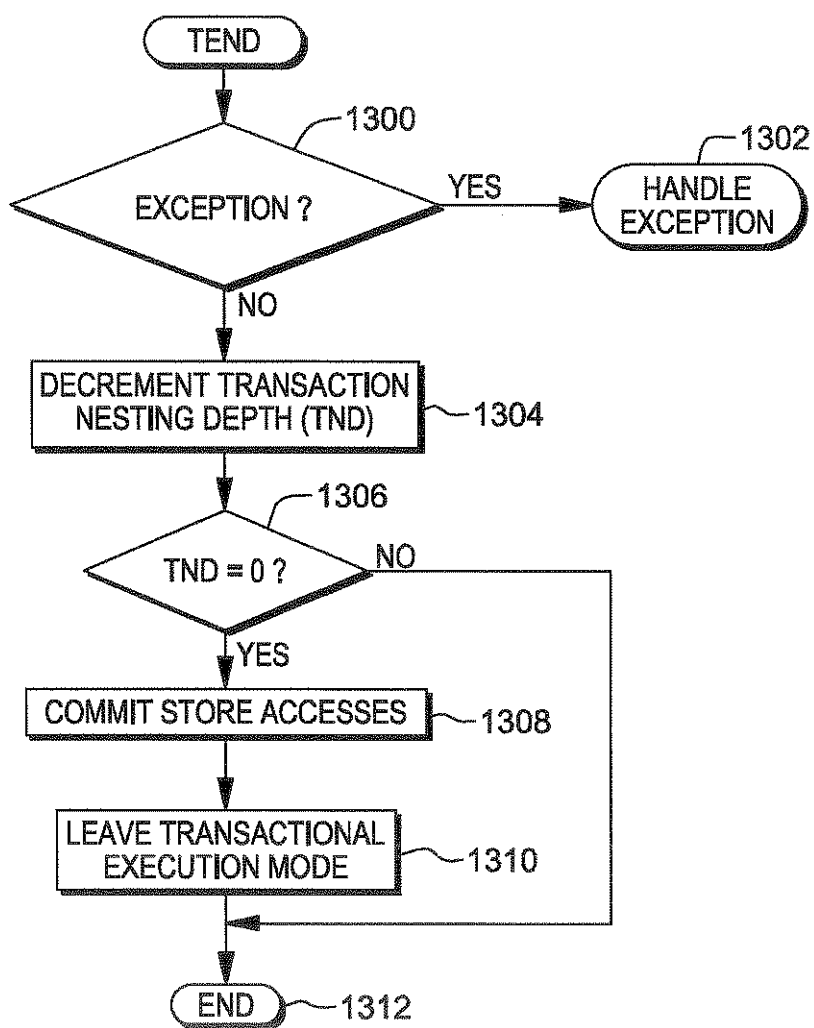


FIG. 13

11/26

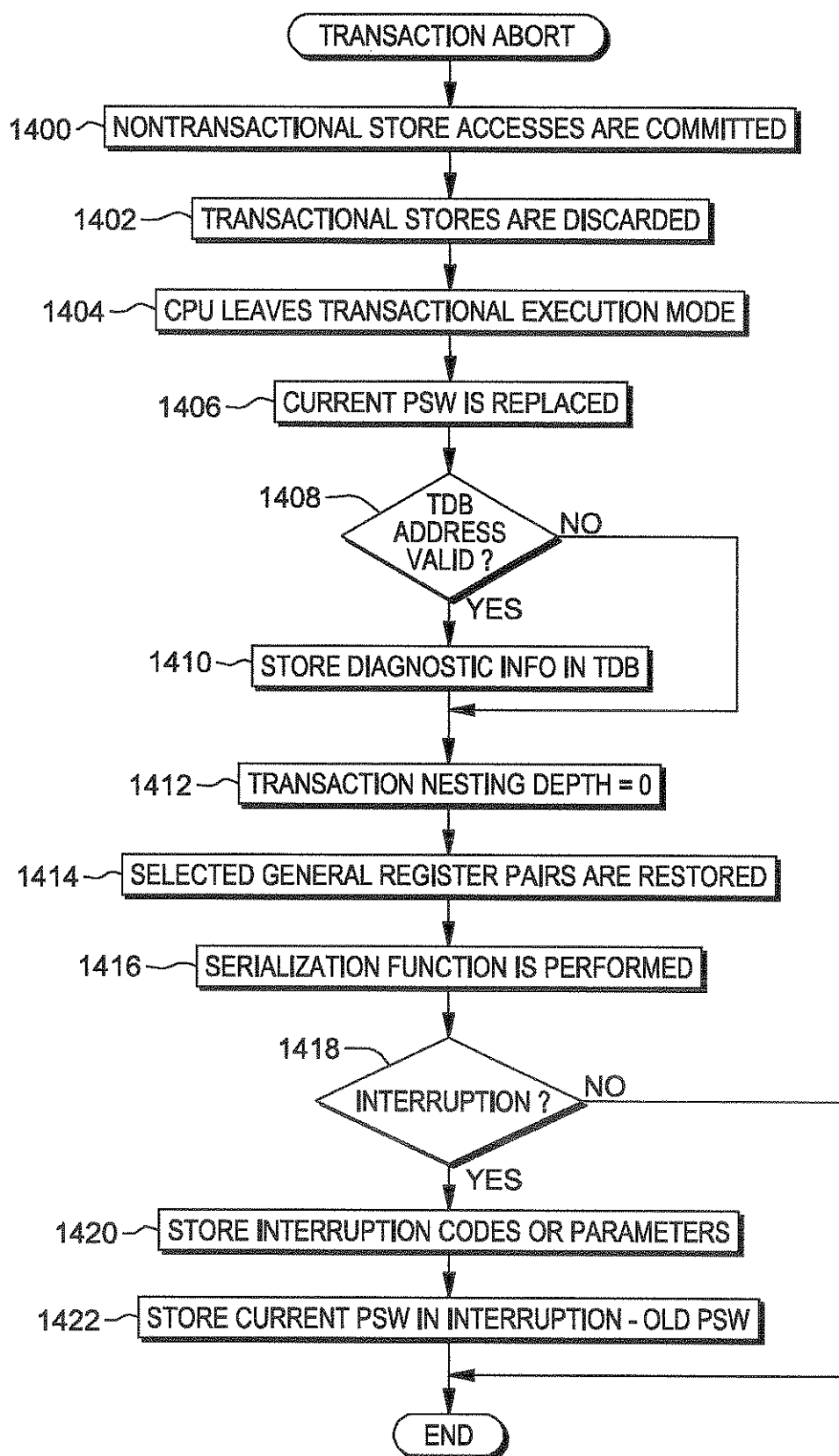


FIG. 14

12/26

Effective PIFC	Program - Interruption Filtering	Result Based on Transactional- Execution (TX) Class		
		1	2	3
0	None	Interrupt	Interrupt	Interrupt
1	Limited	Interrupt	Interrupt	Filtered
2	Moderate	Interrupt	Filtered	Filtered

Explanation:

PIFC Program-Interruption-Filtering Control (bits 14-15 of the I₂ field of the TBEGIN instruction; zeros for TBEGINC); the effective PIFC is the highest value specified (or implied) for the current and all outer-level TRANSACTION BEGIN instructions.

FIG. 15

Code (Hex)	Exception Condition	TX Class	CC Set
0001	Operation	1	3
0002	Privileged operation	1	3
0003	Execute	1	3
0004	Protection	1 or 2 †	2 or 3 ‡
0005	Addressing	1 or 2 †	2 or 3 ‡
0006	Specification	3	2 or 3 ‡
0007	Data (DXC 01, 02, and 03)	1	2
0007	Data (all other DXCs)	3	2 or 3 ‡
0008	Fixed-point overflow	3	2 or 3 ‡
0009	Fixed-point divide	3	2 or 3 ‡
000A	Decimal overflow	3	2 or 3 ‡
000B	Decimal Divide	3	2 or 3 ‡
000C	HFP exponent overflow	3	2 or 3 ‡
000D	HFP exponent underflow	3	2 or 3 ‡
000E	HFP significance	3	2 or 3 ‡
000F	HFP divide	3	2 or 3 ‡
0010	Segment translation	1 or 2 †	2 or 3 ‡
0011	Page translation	1 or 2 †	2 or 3 ‡
0012	Translation specification	1	3
0013	Special operation	1	3
0015	Operand	-	-
0016	Trace table	-	-
0018	Transaction constraint	1	3

FIG. 16A

13/26

Code (Hex)	Exception Condition	TX Class	CC Set
001C	Space switch event	-	-
001D	HFP square root	3	2 or 3 ‡
001F	PC-translation specification	-	-
0020	AFX translation	-	-
0021	ASX translation	-	-
0022	LX translation	-	-
0023	EX translation	-	-
0024	Primary authority	-	-
0025	Secondary authority	-	-
0026	LFX translation	-	-
0027	LSX translation	-	-
0028	ALET specification	2	2 or 3 ‡
0029	ALEN translation	2	2 or 3 ‡
002A	ALE sequence	2	2 or 3 ‡
002B	ASTE validity	2	2 or 3 ‡
002C	ASTE sequence	2	2 or 3 ‡
002D	Extended authority	2	2 or 3 ‡
002E	LSTE sequence	-	-
002F	ASTE instance	-	-
0030	Stack full	-	-
0031	Stack empty	-	-
0032	Stack specification	-	-
0033	Stack type	-	-
0034	Stack operation	-	-
0038	ASCE type	1 or 2 †	2 or 3 ‡
0039	Region-first translation	1 or 2 †	2 or 3 ‡
003A	Region-second translation	1 or 2 †	2 or 3 ‡
003B	Region-third translation	1 or 2 †	2 or 3 ‡
0040	Monitor event	-	-
0080	PER event	1	3 ★

Explanation:

- Not applicable; the exception cannot occur in the transactional-execution mode because the instruction that causes the exception is a restricted instruction.
- † Access exceptions recognized during fetching of an instruction in the transactional execution mode are TX class 1 (that is, they cannot be filtered). Access exceptions when accessing a storage operand in transactional-execution mode are TX class 2.
- ‡ When the exception condition is not filtered, the condition code is 2; when the condition is filtered, the condition code is 3.
- ★ The CPU sets condition code 3. A control program or hypervisor may change the condition code in the program-old PSW to 2.

FIG. 16B

14/26

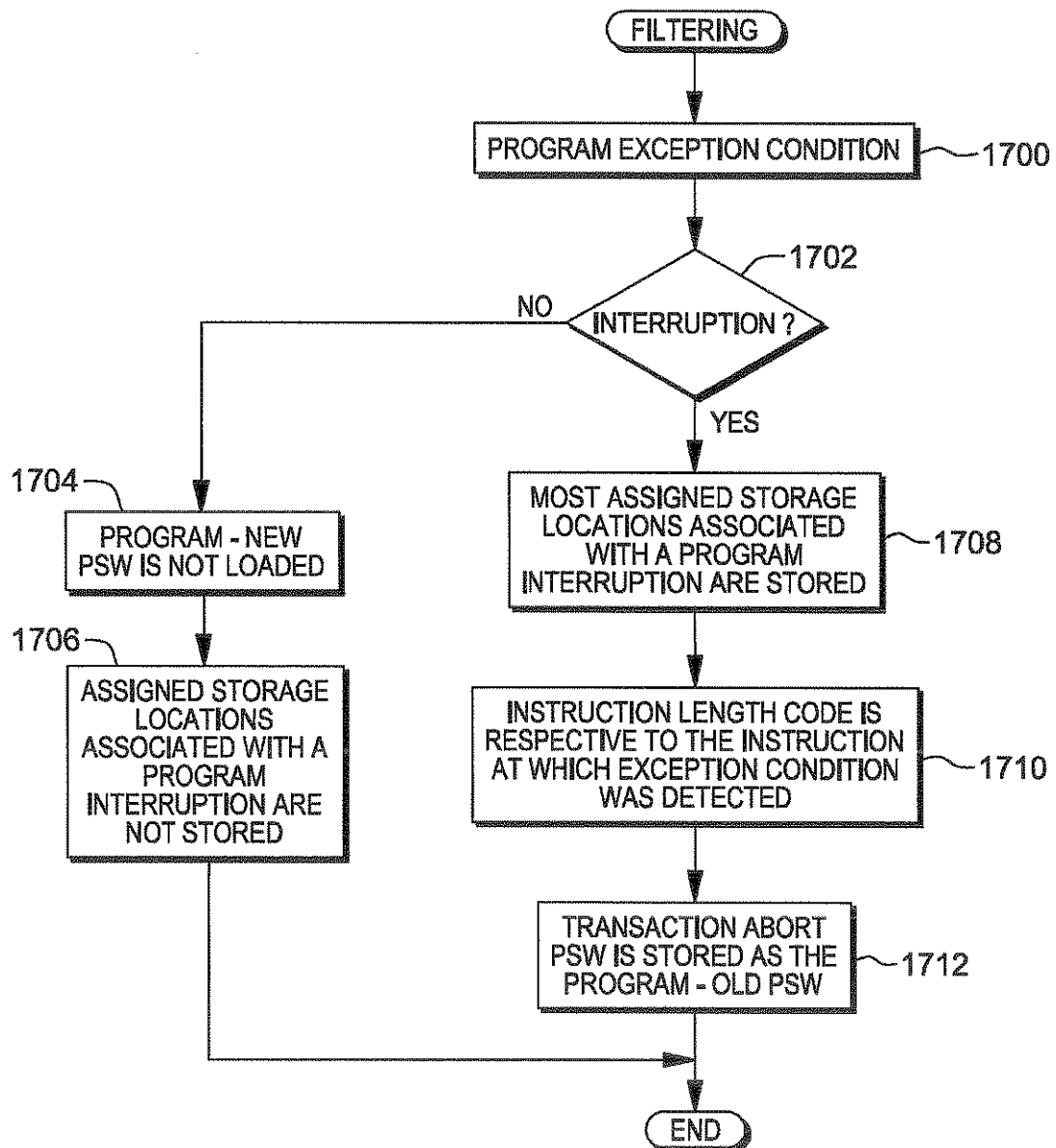


FIG. 17A

15/26

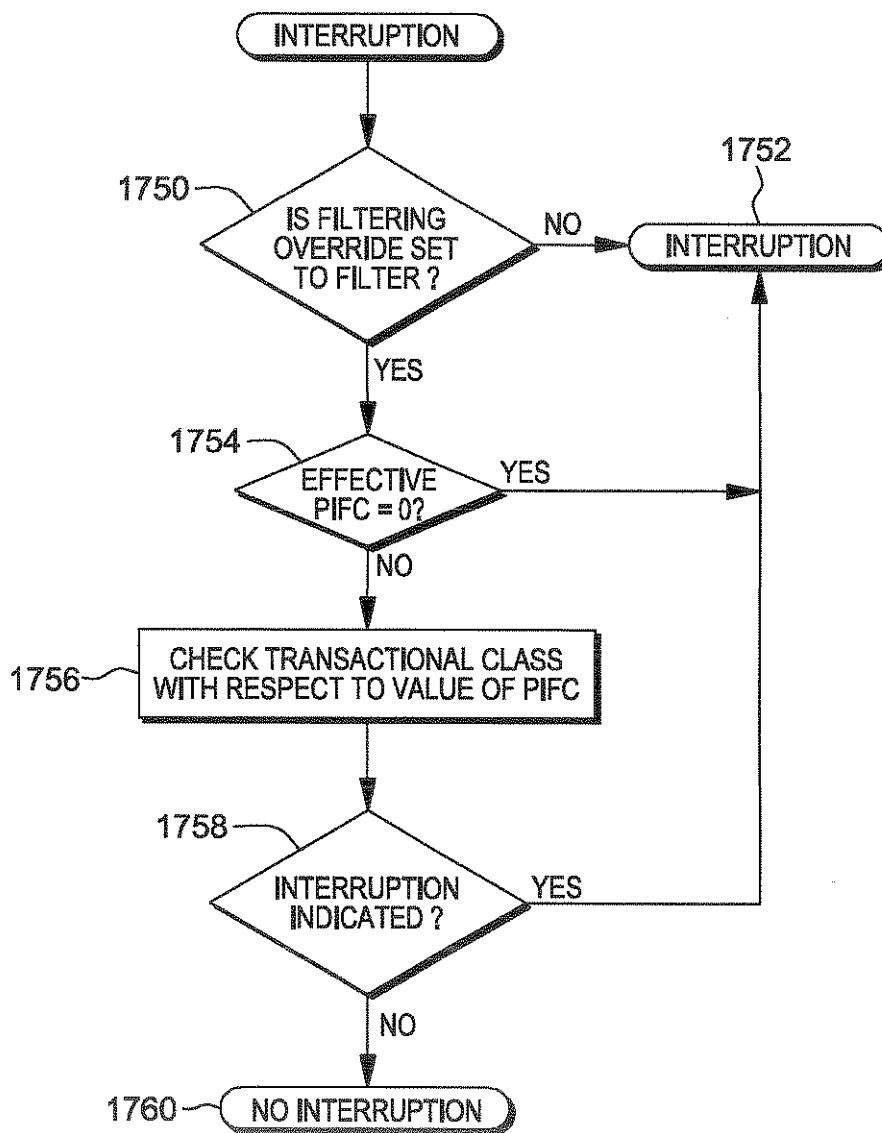


FIG. 17B

INSERTING AN ELEMENT INTO A DOUBLY-LINKED LIST (BEFORE)

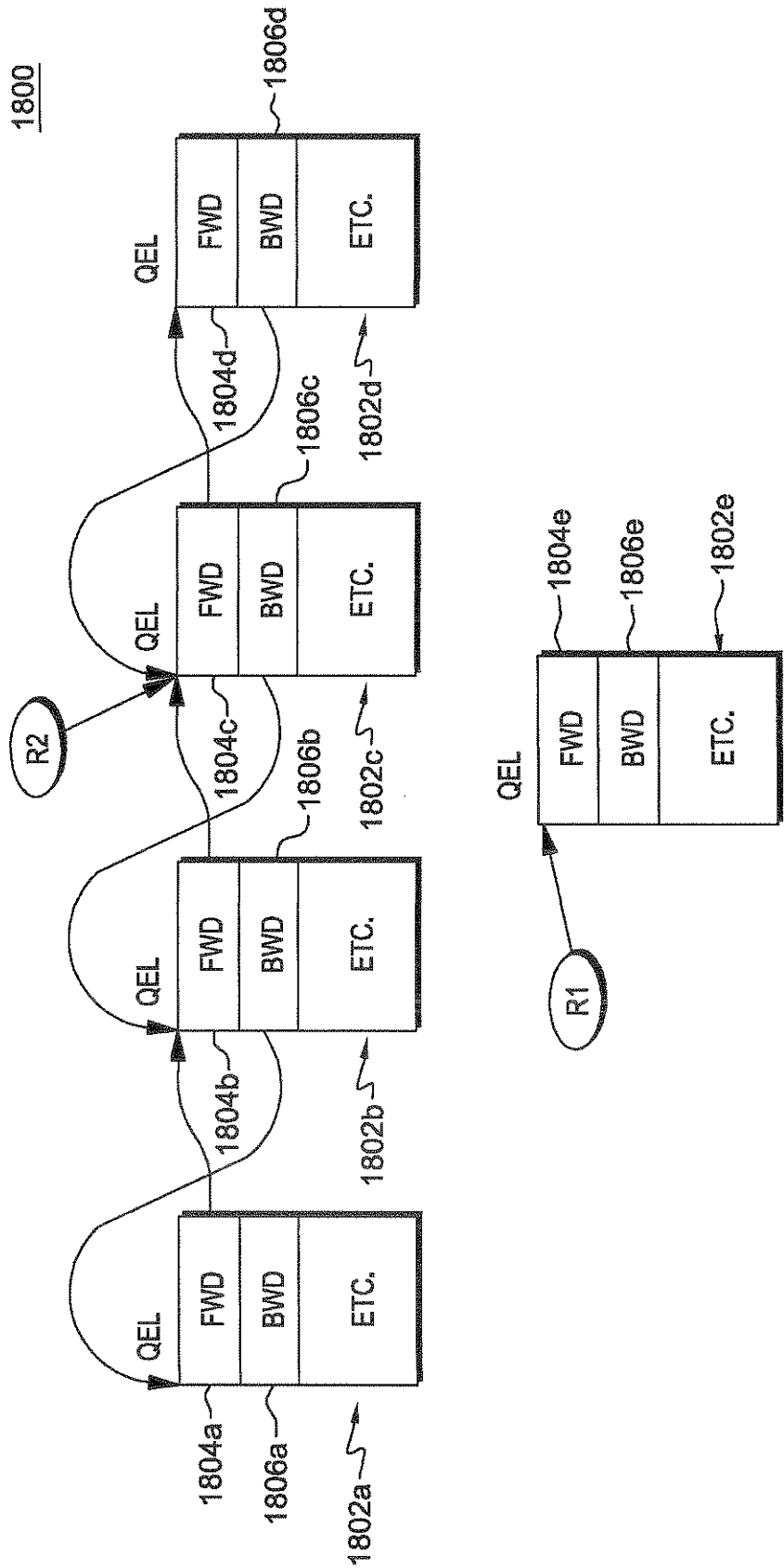
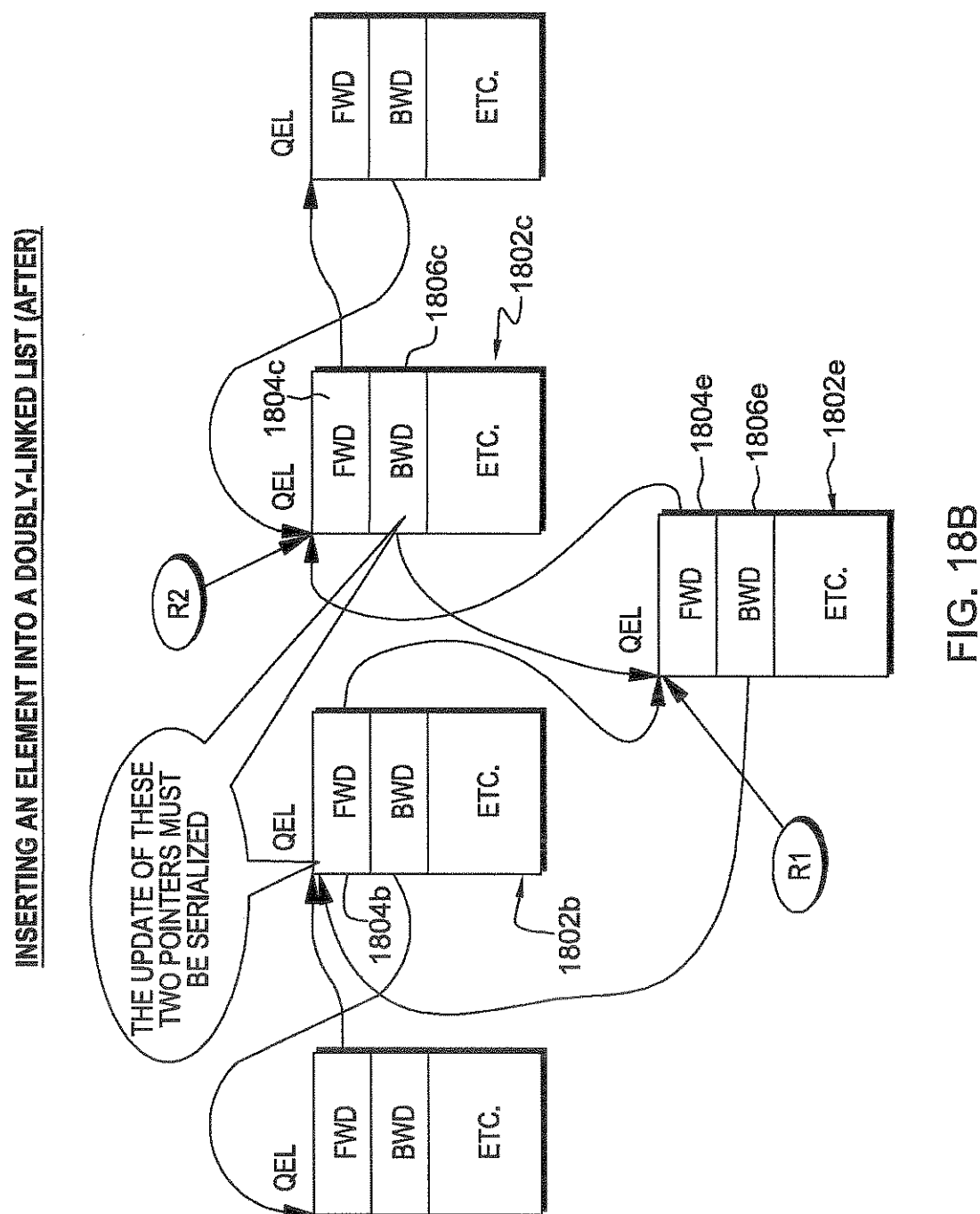


FIG. 18A



18/26

COMPUTER
PROGRAM
PRODUCT
1900

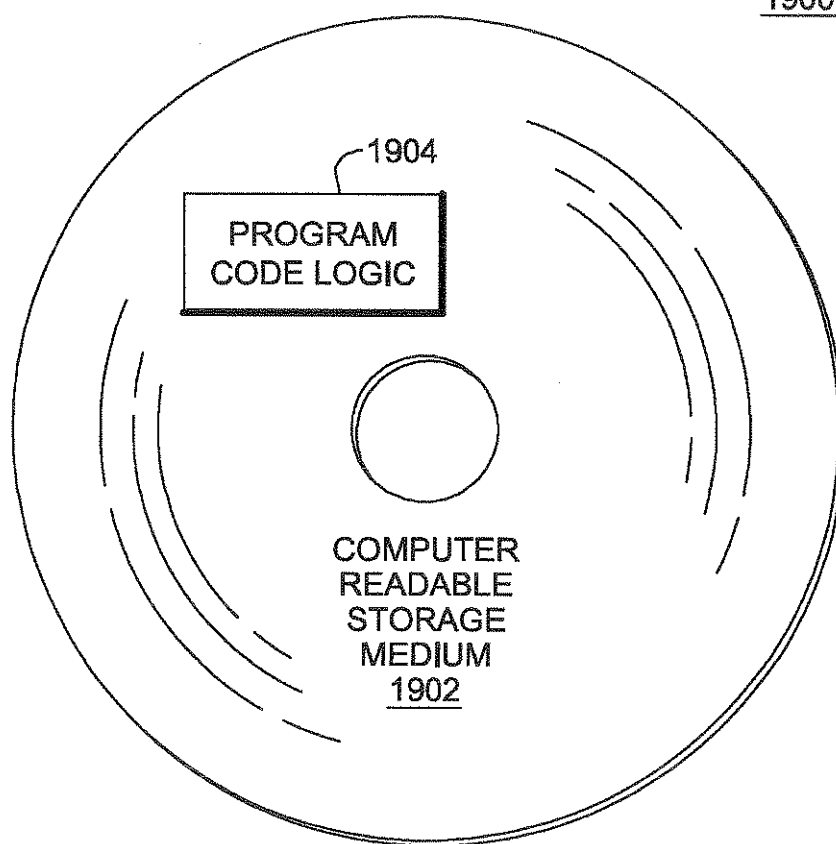
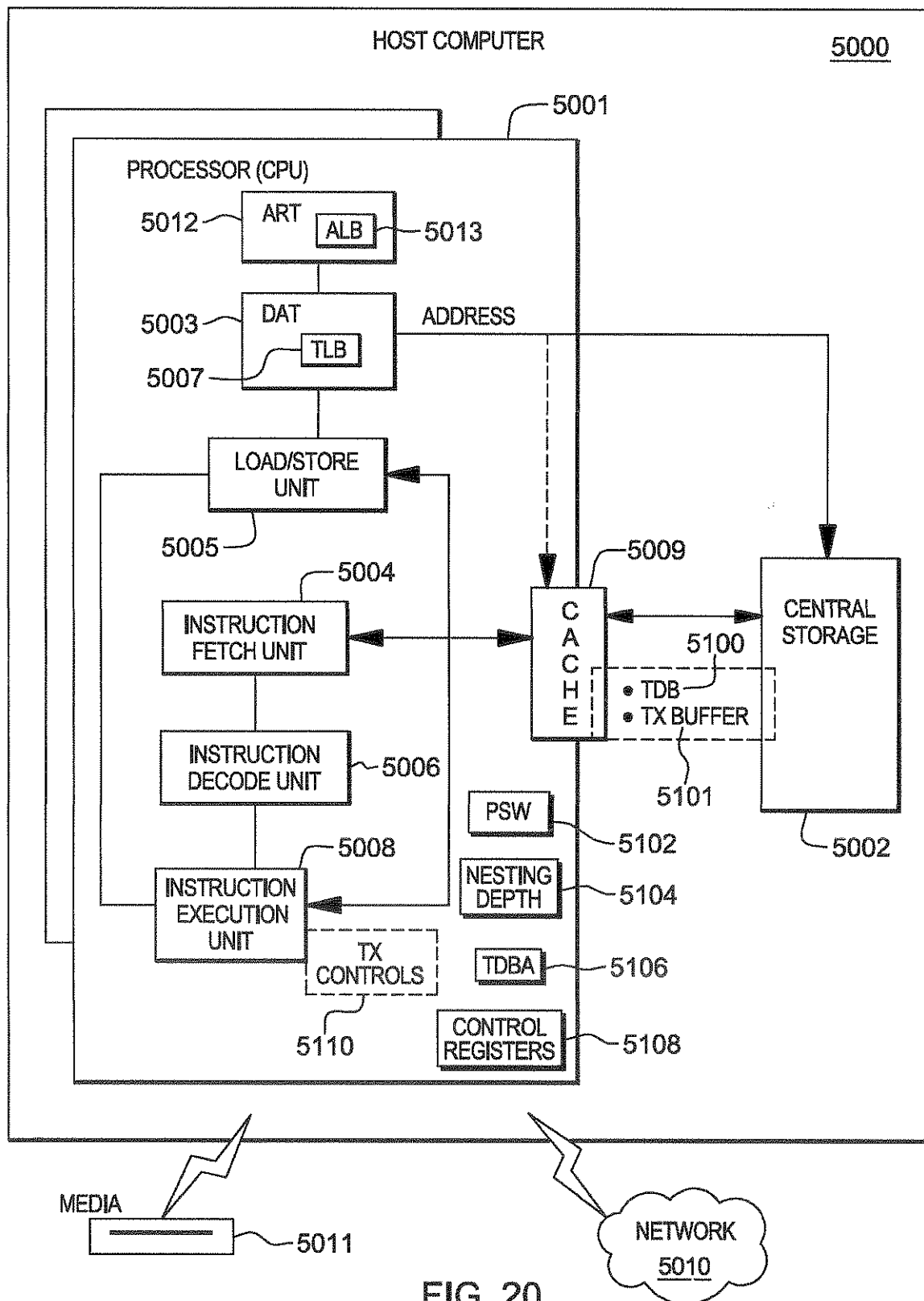


FIG. 19

19/26



20/26

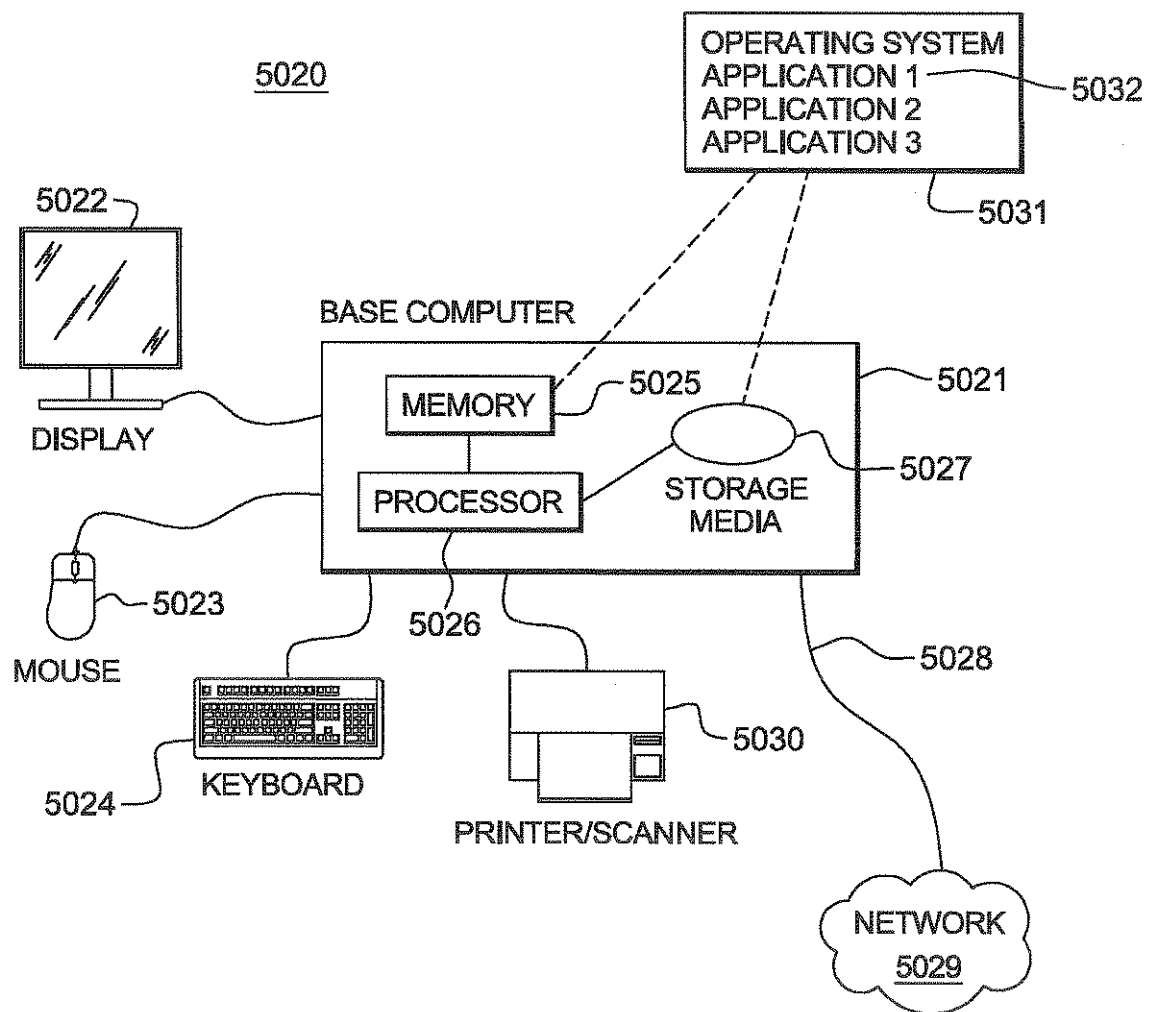


FIG. 21

21/26

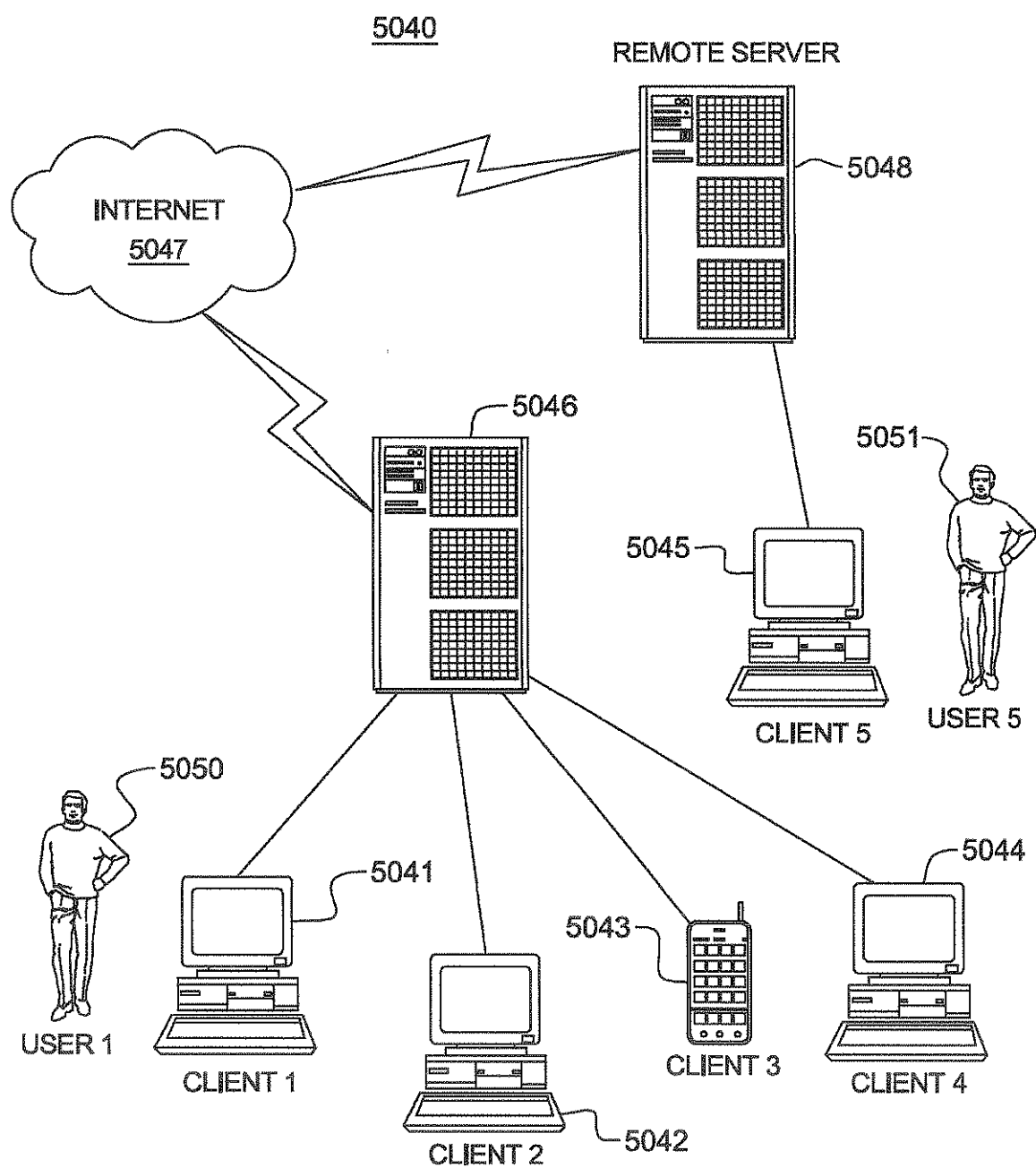


FIG. 22

22/26

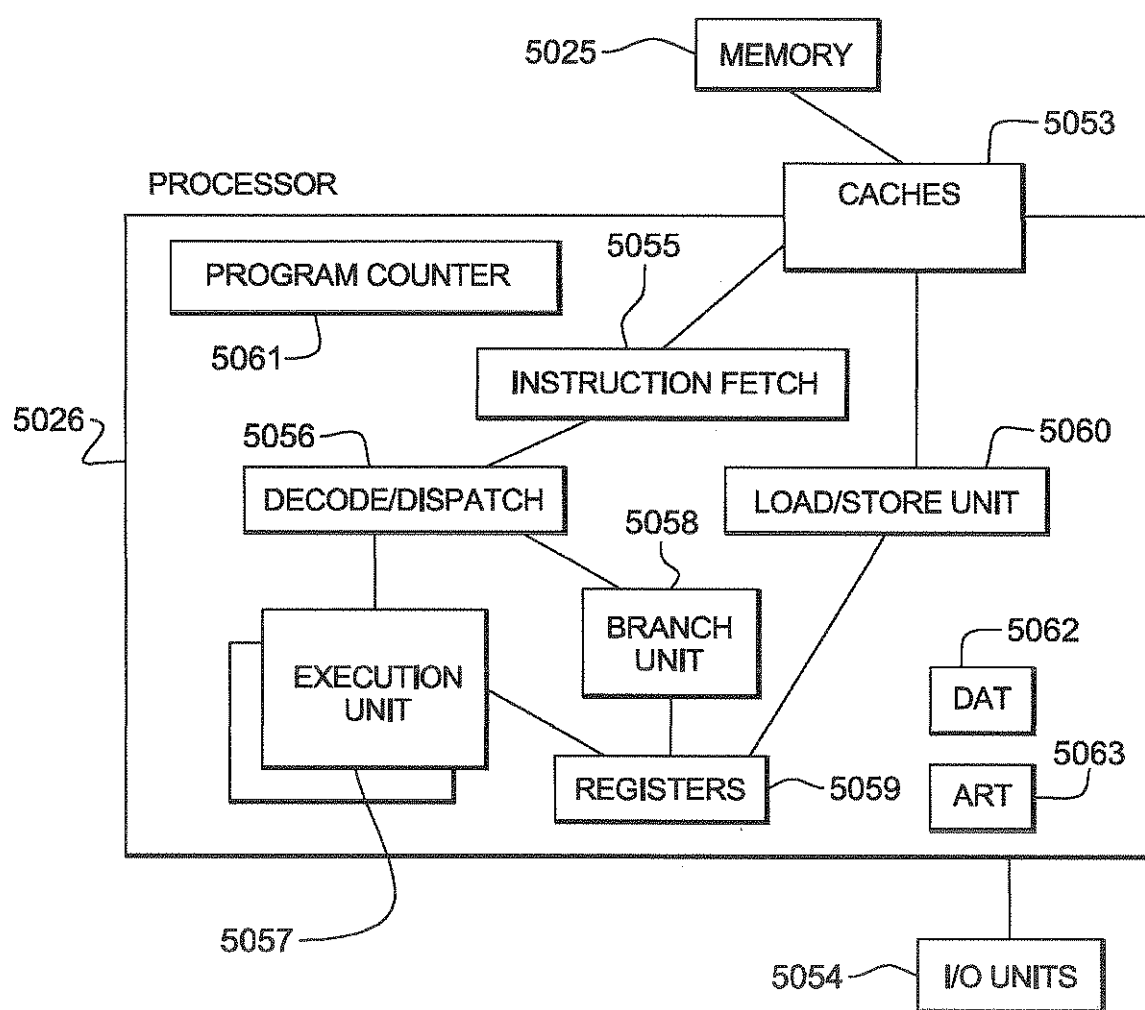


FIG. 23

23/26

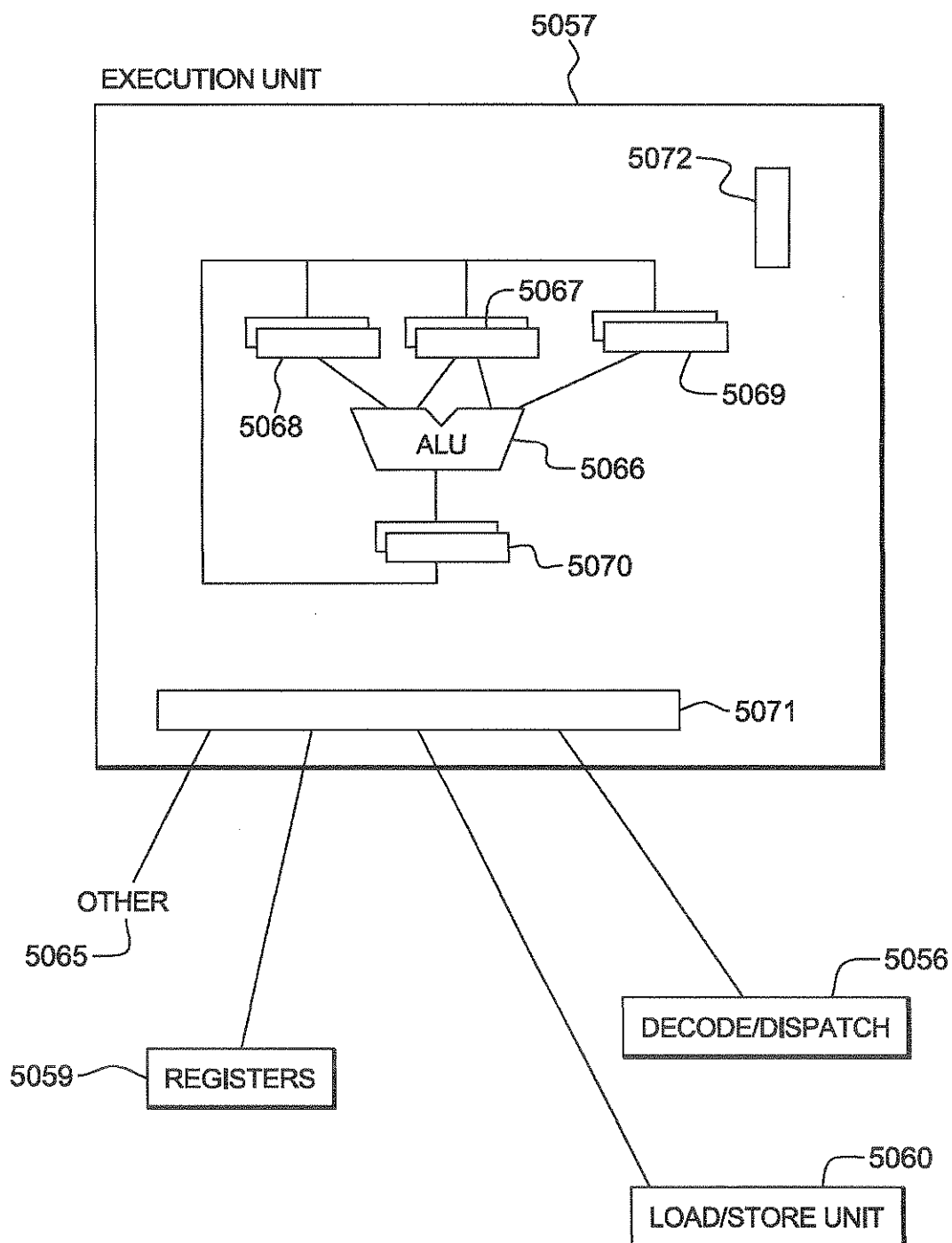


FIG. 24A

24/26

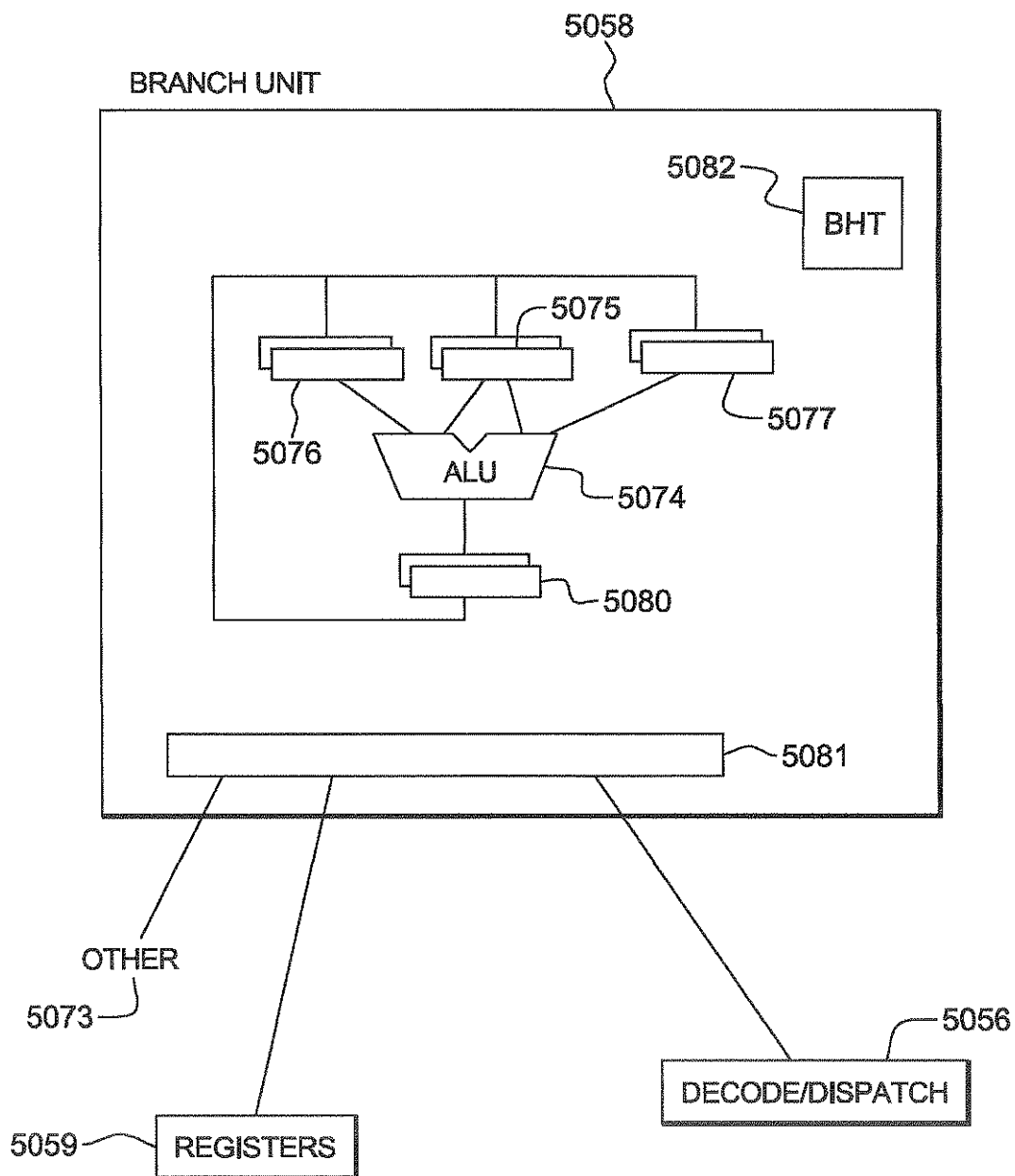


FIG. 24B

25/26

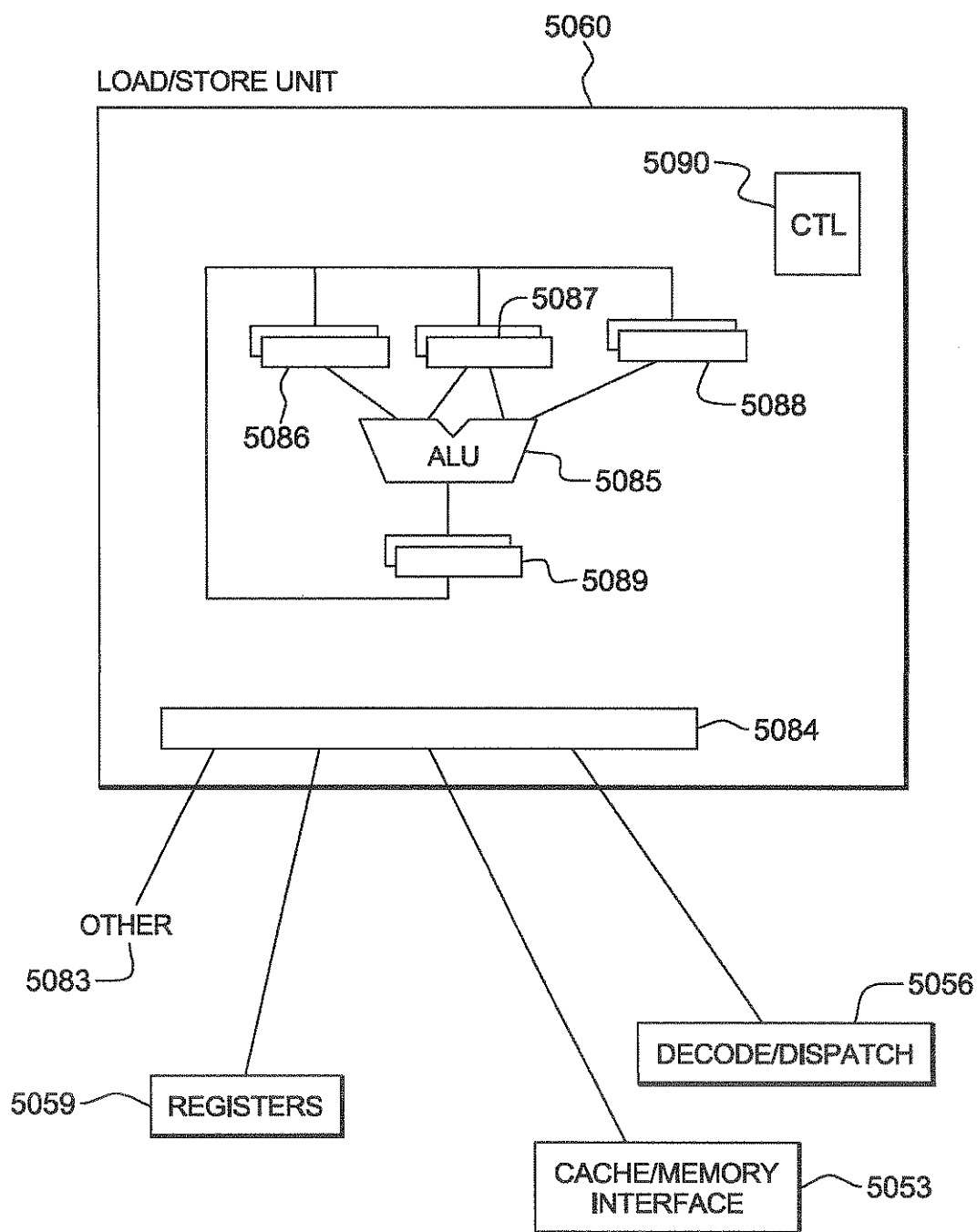


FIG. 24C

26/26

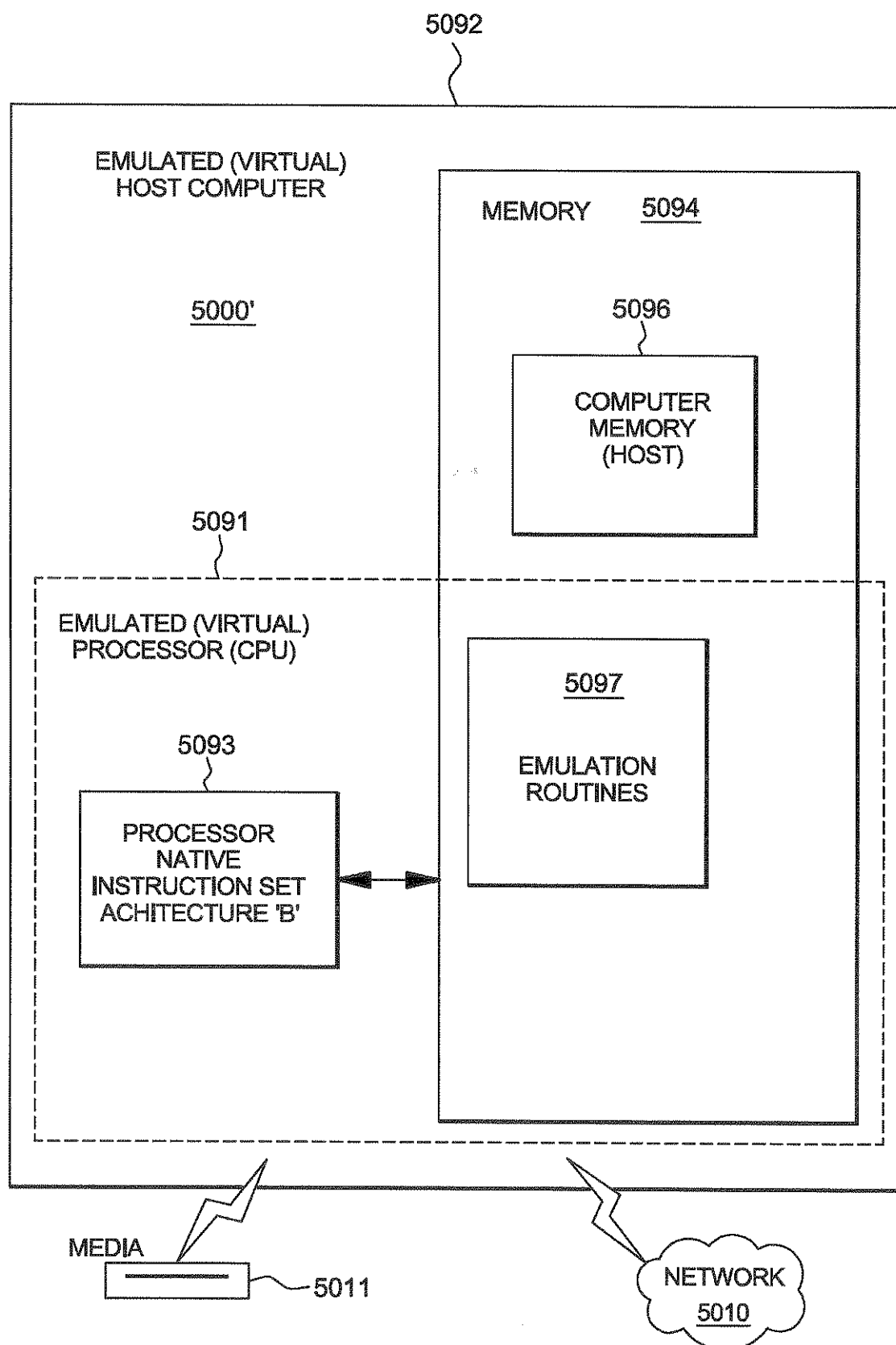


FIG. 25