United States Patent [19]

Mandel

[11] **3,807,531**

[45] **Apr. 30, 1974**

[54] **ELEVATOR SYSTEM**

[75] Inventor: **Alan F. Mandel**, Pittsburgh, Pa.

[73] Assignee: **Westinghouse Electric Corporation**, Pittsburgh, Pa.

[22] Filed: **Mar. 12, 1973**

[21] Appl. No.: **340,621**

[52] **U.S. Cl.** ..... **187/29 R**, 340/19 R

[51] **Int. Cl.** **B66b 3/00**

[58] **Field of Search** 187/29; 340/19–21

[56] **References Cited**

UNITED STATES PATENTS

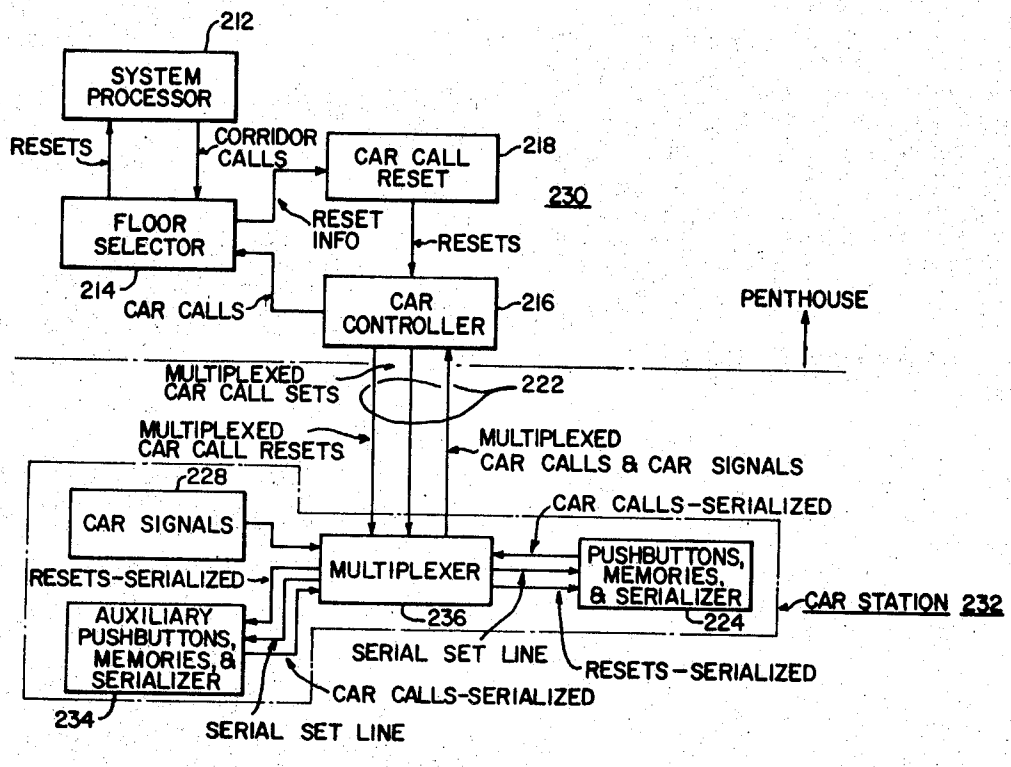
3,353,631	11/1967	Burgy	187/29
2,738,489	3/1956	Borden	340/19

Primary Examiner—Bernard A. Gilheany
Assistant Examiner—W. E. Duncanson, Jr.
Attorney, Agent, or Firm—D. R. Lackey

[57] **ABSTRACT**

An elevator system including an elevator car having main and auxiliary car stations, each of which include pushbuttons for registering calls for service to various landings of an associated structure. The car calls are stored and serialized in the main and auxiliary car stations, and a single serial car call signal is provided in response to the serial signals from the main and auxiliary car stations. A serial set line is connected to each car station responsive to car calls originating at the other car station, simulating parallel interconnection of the pushbuttons of the two car stations by setting a call on one station which originated at the other station.

8 Claims, 42 Drawing Figures



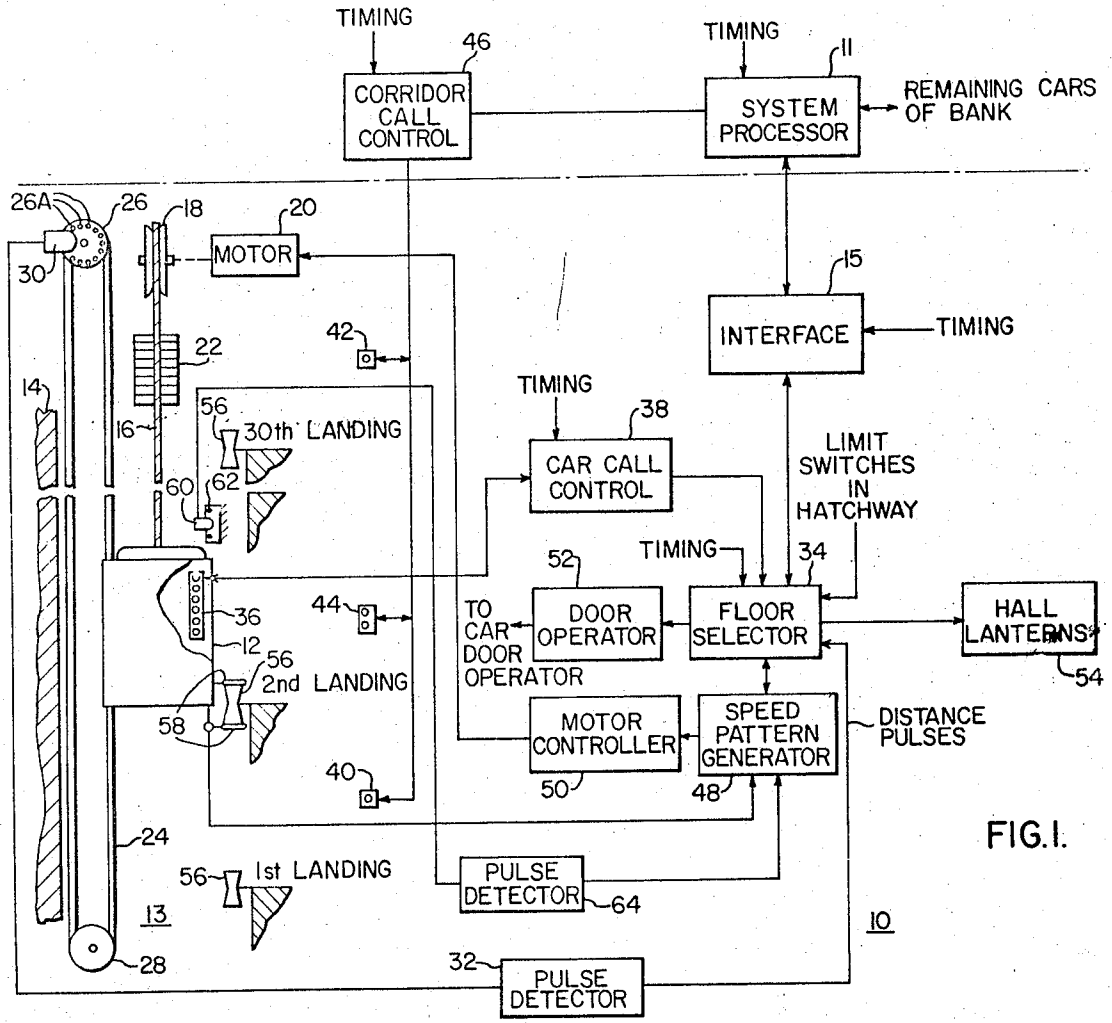
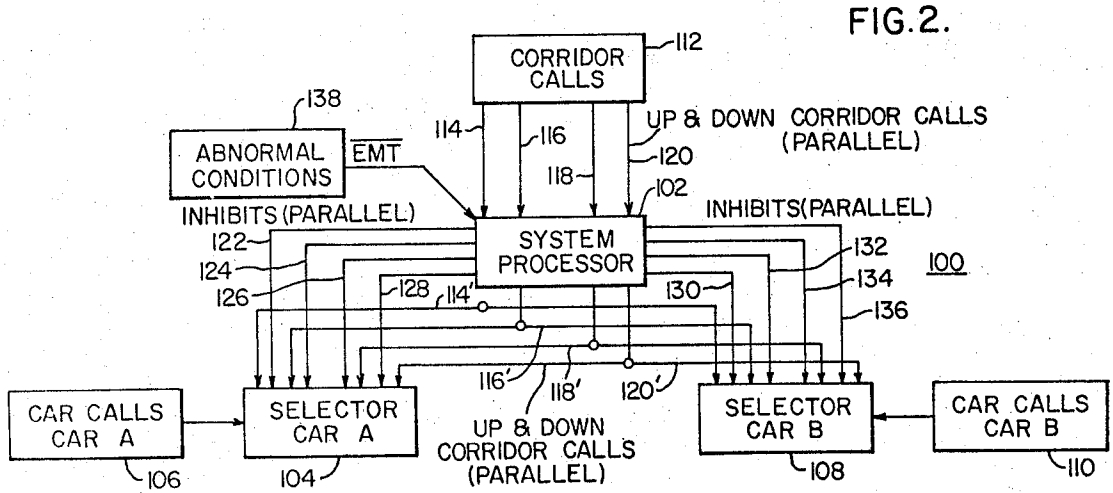

TIMING
46
CORRIDOR CALL CONTROL
TIMING
11
SYSTEM PROCESSOR
REMAINING CARS OF BANK
26A
26
18
20
MOTOR
30
15
INTERFACE
TIMING
22
42
14
TIMING
38
56
30th LANDING
16
CAR CALL CONTROL
LIMIT SWITCHES IN HATCHWAY
60
62
TIMING
34
52
36
44
TO CAR DOOR OPERATOR
DOOR OPERATOR
FLOOR SELECTOR
HALL LANTERNS
54
12
56
2nd LANDING
58
MOTOR CONTROLLER
SPEED PATTERN GENERATOR
DISTANCE PULSES
40
50
48
24
56
1st LANDING
PULSE DETECTOR
64
13
10
28
32
PULSE DETECTOR

FIG.1.

FIG.2.


CORRIDOR CALLS
112
138
ABNORMAL CONDITIONS
EMT
114
116
118
120
UP & DOWN CORRIDOR CALLS (PARALLEL)
102
INHIBITS (PARALLEL)
INHIBITS(PARALLEL)
122
124
126
SYSTEM PROCESSOR
132
134
136
100
128
114'
130
116'
118'
120'
CAR CALLS CAR A
SELECTOR CAR A
UP & DOWN CORRIDOR CALLS (PARALLEL)
SELECTOR CAR B
CAR CALLS CAR B
106
104
108
110

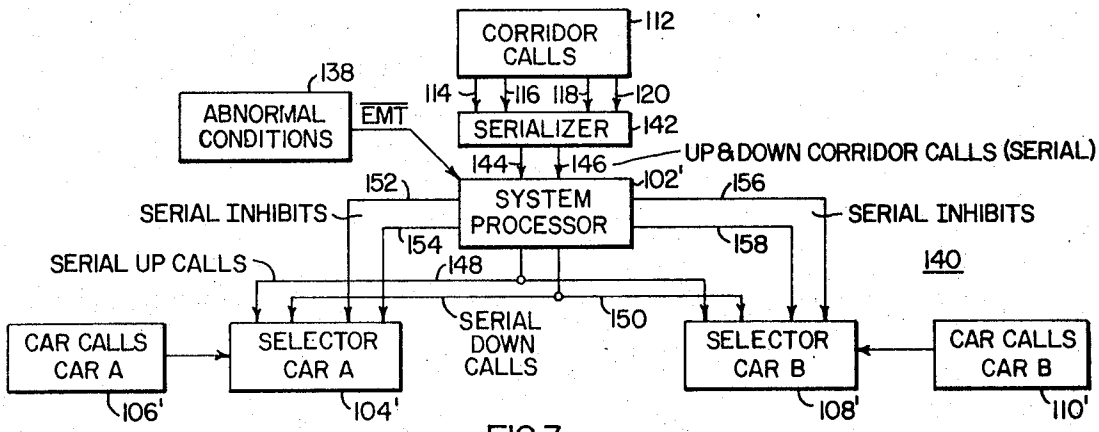

CORRIDOR CALLS
112
138
ABNORMAL CONDITIONS
EMT
114
116
118
120
SERIALIZER
142
144
146
UP & DOWN CORRIDOR CALLS (SERIAL)
152
SYSTEM PROCESSOR
102'
156
SERIAL INHIBITS
SERIAL INHIBITS
154
158
140
SERIAL UP CALLS
148
SERIAL DOWN CALLS
150
CAR CALLS CAR A
SELECTOR CAR A
SELECTOR CAR B
CAR CALLS CAR B
106'
104'
108'
110'

FIG.3

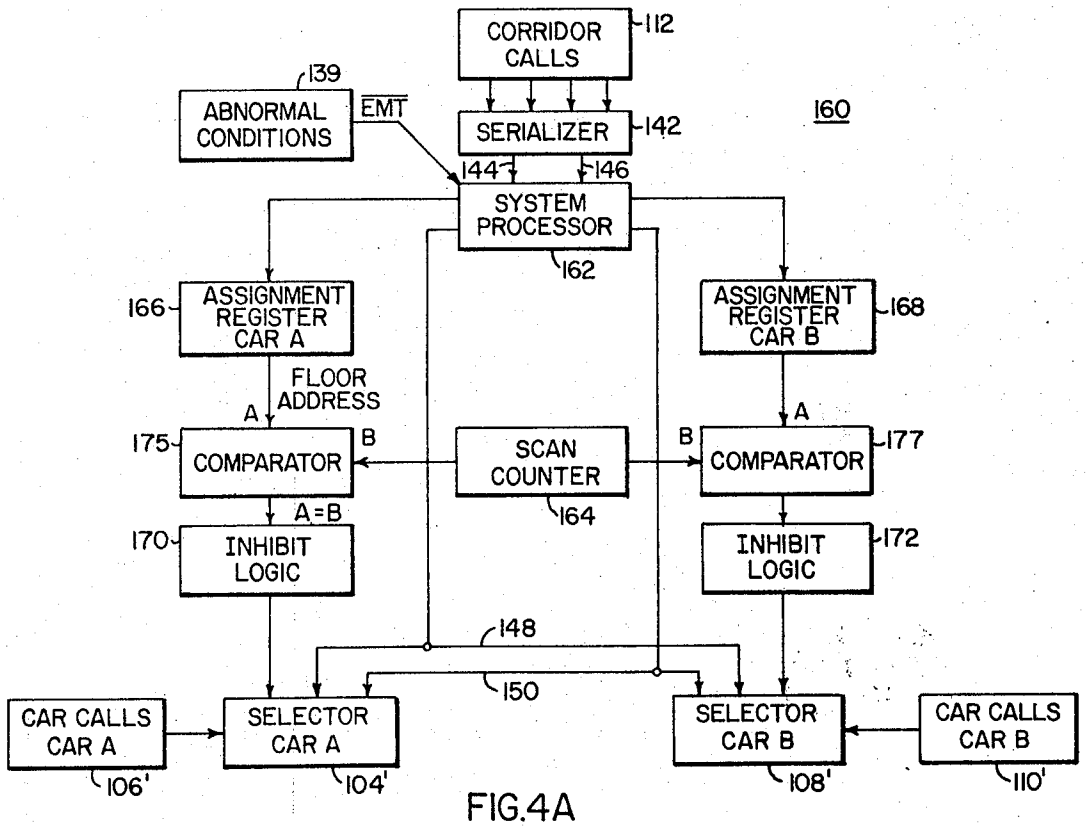

CORRIDOR CALLS
112
139
ABNORMAL CONDITIONS
EMT
SERIALIZER
142
160
144
146
SYSTEM PROCESSOR
162
166
ASSIGNMENT REGISTER CAR A
ASSIGNMENT REGISTER CAR B
168
FLOOR ADDRESS
A
B
175
COMPARATOR
SCAN COUNTER
COMPARATOR
177
A=B
170
INHIBIT LOGIC
164
INHIBIT LOGIC
172
148
150
CAR CALLS CAR A
SELECTOR CAR A
SELECTOR CAR B
CAR CALLS CAR B
106'
104'
108'
110'

FIG.4A

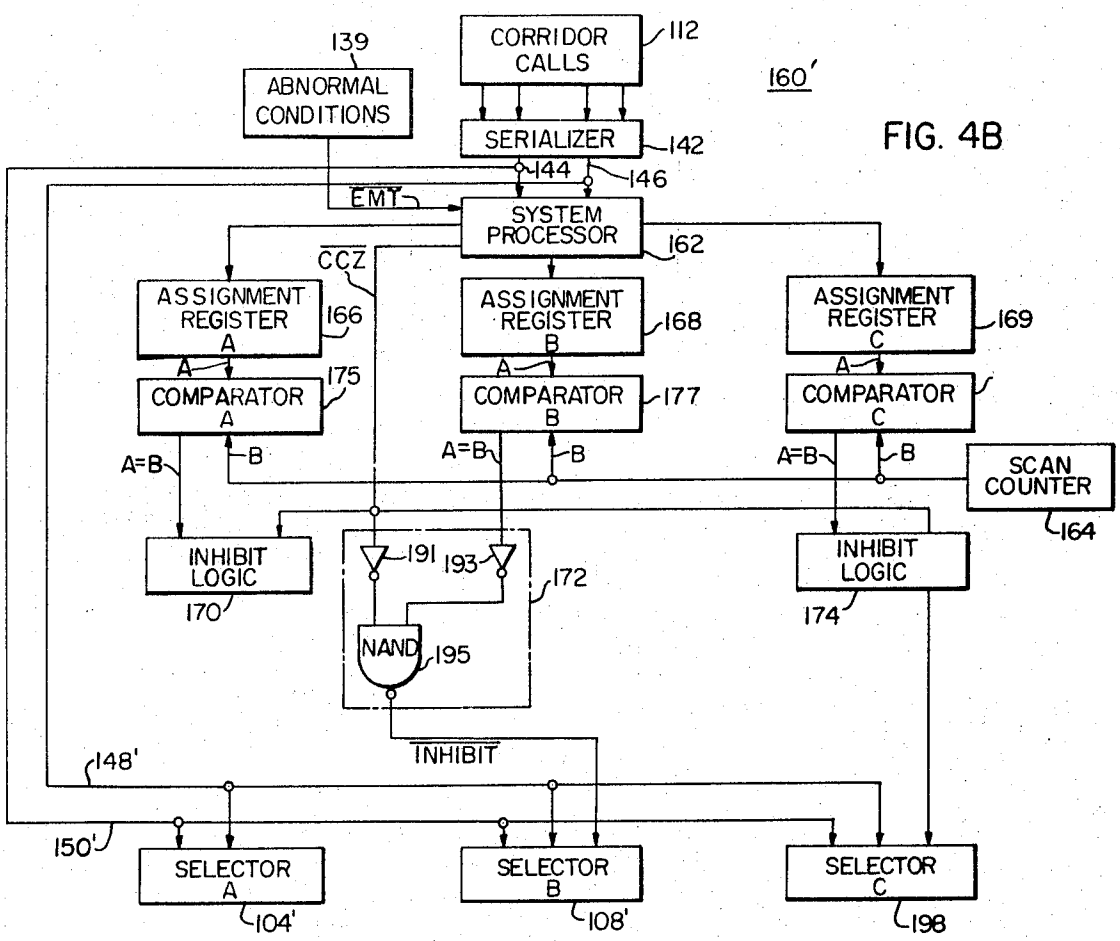

139
ABNORMAL CONDITIONS
CORRIDOR CALLS
112
160'
SERIALIZER
142
144
146
EMT
SYSTEM PROCESSOR
162
CCZ
ASSIGNMENT REGISTER A
166
ASSIGNMENT REGISTER B
168
ASSIGNMENT REGISTER C
169
A
COMPARATOR A
175
COMPARATOR B
177
COMPARATOR C
A=B
B
SCAN COUNTER
164
INHIBIT LOGIC
170
191
193
172
NAND
195
INHIBIT LOGIC
174
INHIBIT
148'
150'
SELECTOR A
104'
SELECTOR B
108'
SELECTOR C
198

FIG. 4B

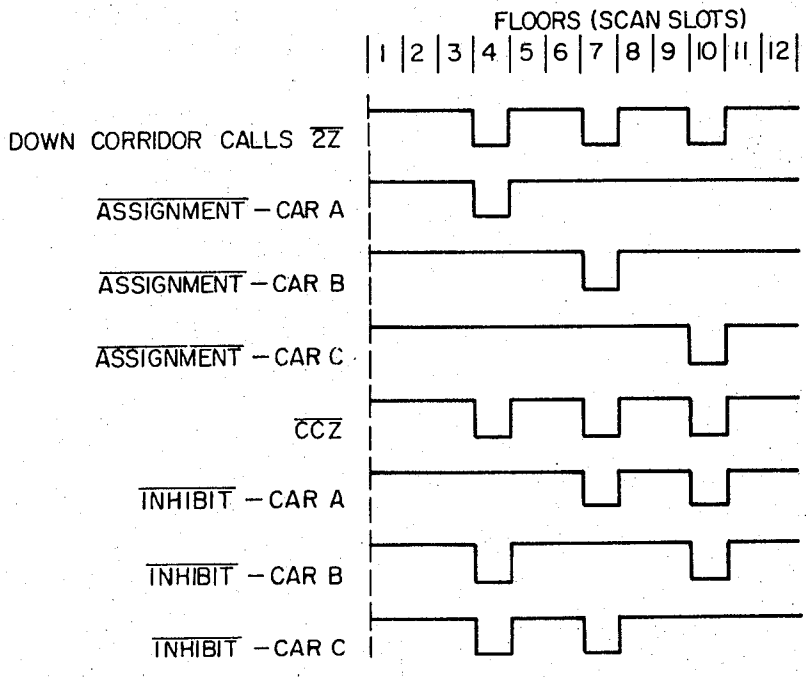

FLOORS (SCAN SLOTS)
1 2 3 4 5 6 7 8 9 10 11 12
DOWN CORRIDOR CALLS ZZ
ASSIGNMENT – CAR A
ASSIGNMENT – CAR B
ASSIGNMENT – CAR C
CCZ
INHIBIT – CAR A
INHIBIT – CAR B
INHIBIT – CAR C

FIG. 4C

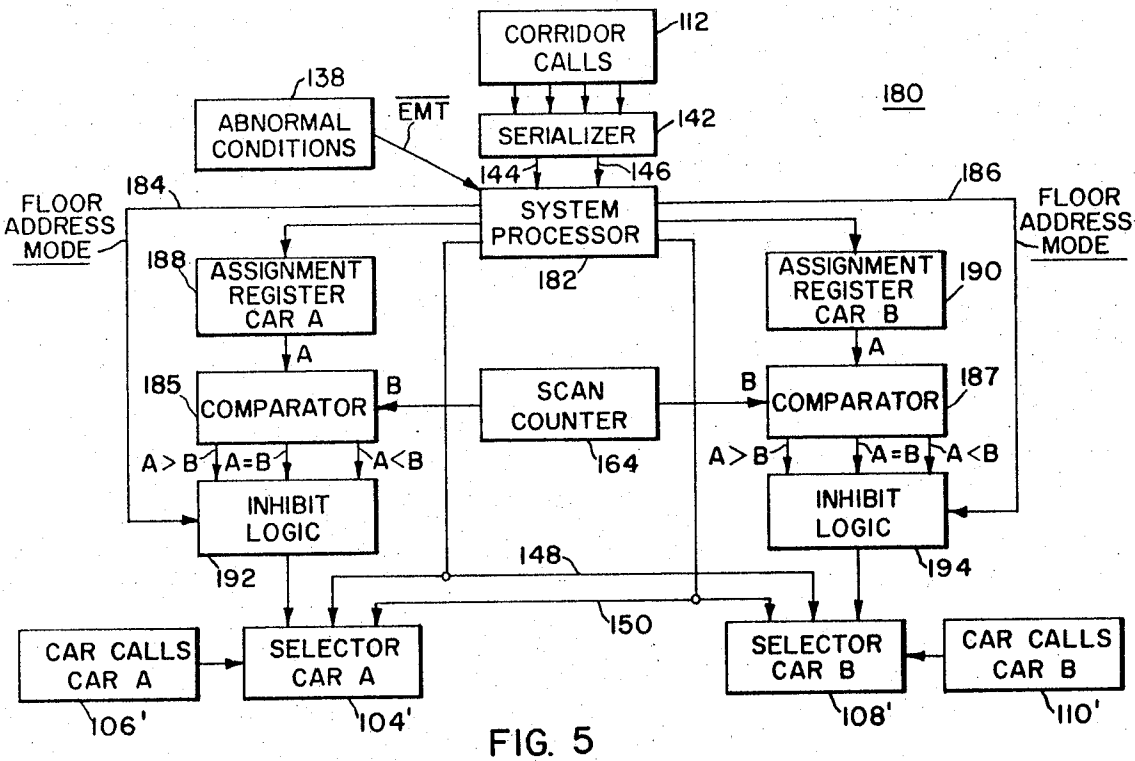

CORRIDOR CALLS
112
180
138
ABNORMAL CONDITIONS
EMT
SERIALIZER
142
144
146
186
184
FLOOR ADDRESS MODE
SYSTEM PROCESSOR
182
FLOOR ADDRESS MODE
188
ASSIGNMENT REGISTER CAR A
ASSIGNMENT REGISTER CAR B
190
A
185
COMPARATOR
B
SCAN COUNTER
164
B
COMPARATOR
187
A>B
A=B
A<B
A>B
A=B
A<B
INHIBIT LOGIC
192
INHIBIT LOGIC
194
148
150
CAR CALLS CAR A
106'
SELECTOR CAR A
104'
SELECTOR CAR B
108'
CAR CALLS CAR B
110'

FIG. 5

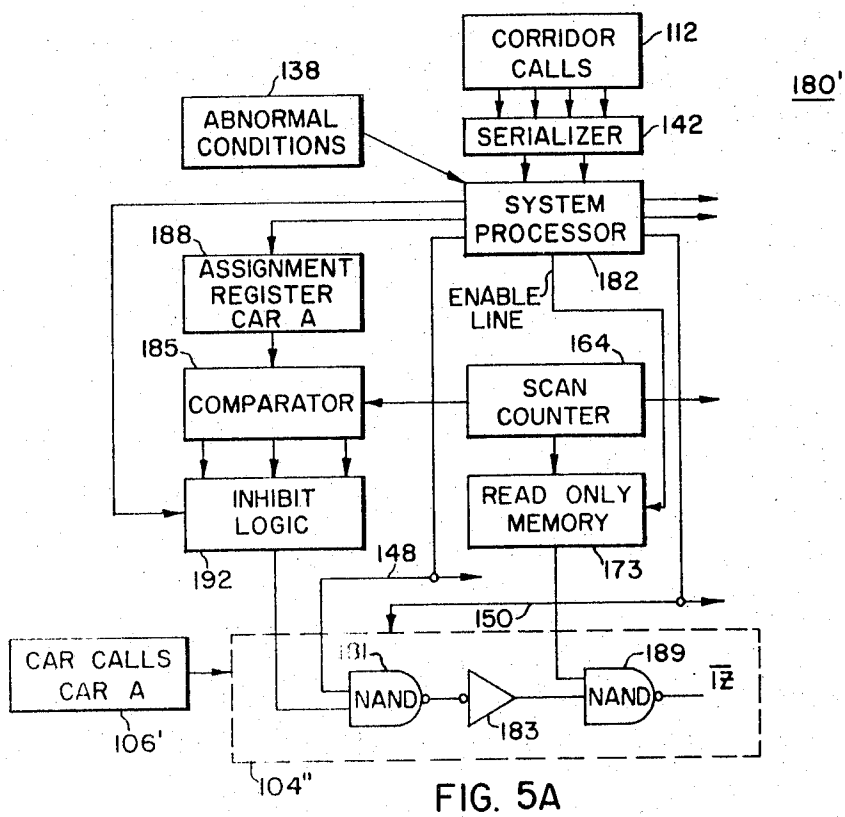

CORRIDOR CALLS
112
180'
138
ABNORMAL CONDITIONS
SERIALIZER
142
SYSTEM PROCESSOR
188
ASSIGNMENT REGISTER CAR A
ENABLE LINE
182
185
164
COMPARATOR
SCAN COUNTER
INHIBIT LOGIC
READ ONLY MEMORY
192
148
173
150
CAR CALLS CAR A
106'
131
NAND
183
189
NAND
104"

FIG. 5A

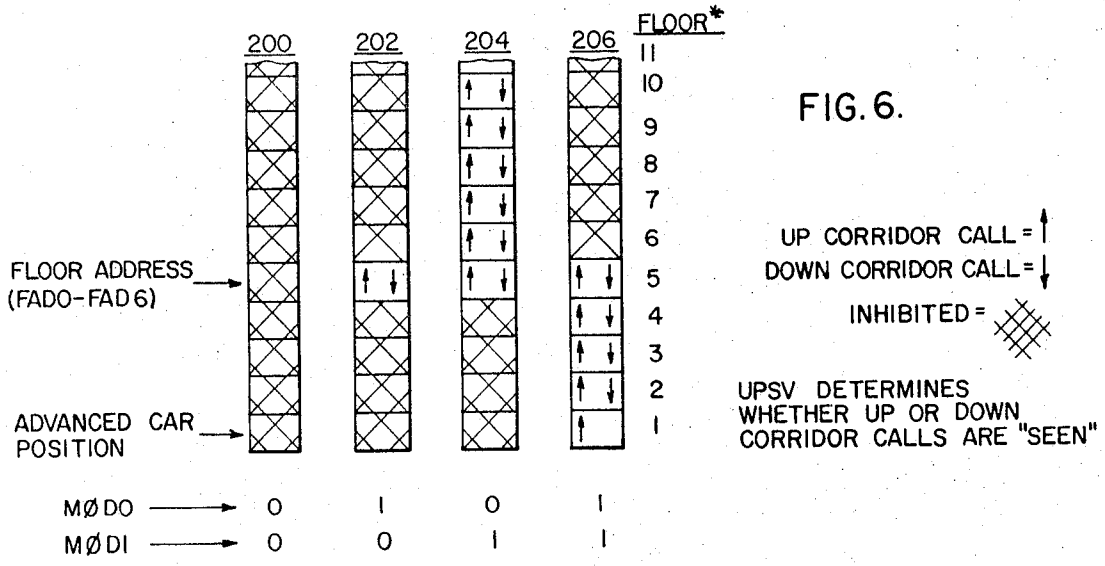


FIG. 6.

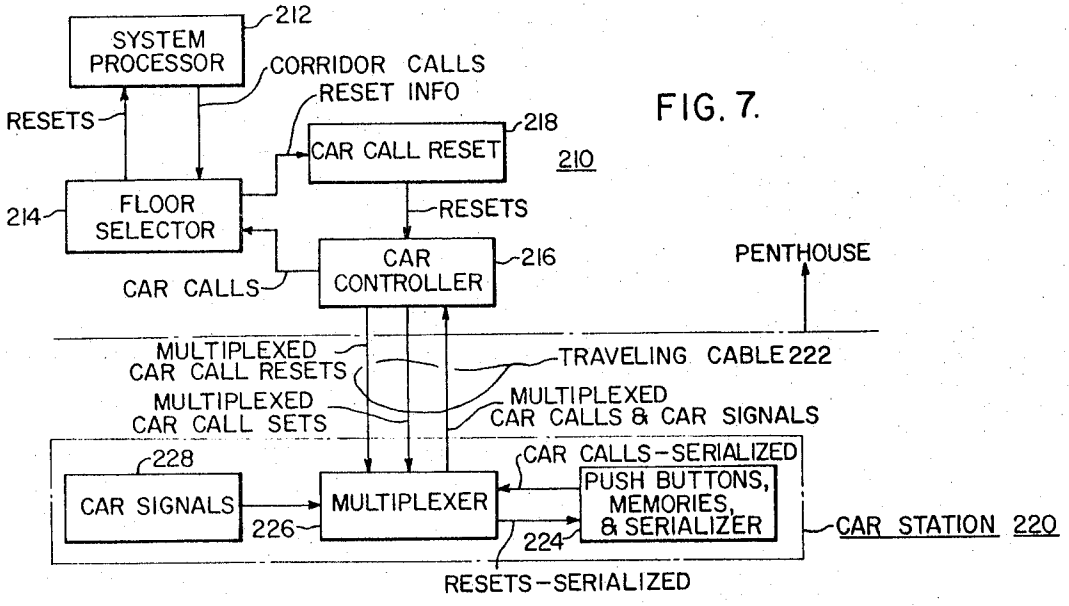


FIG. 7.

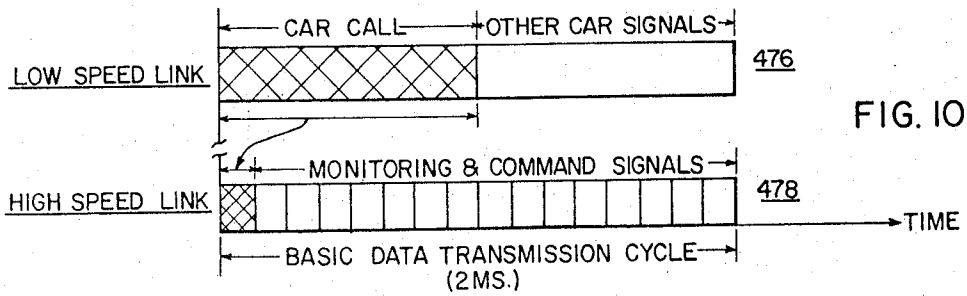


FIG. 10.

TIMING DIAGRAM-DUAL DATA TRANSMISSION SYSTEM

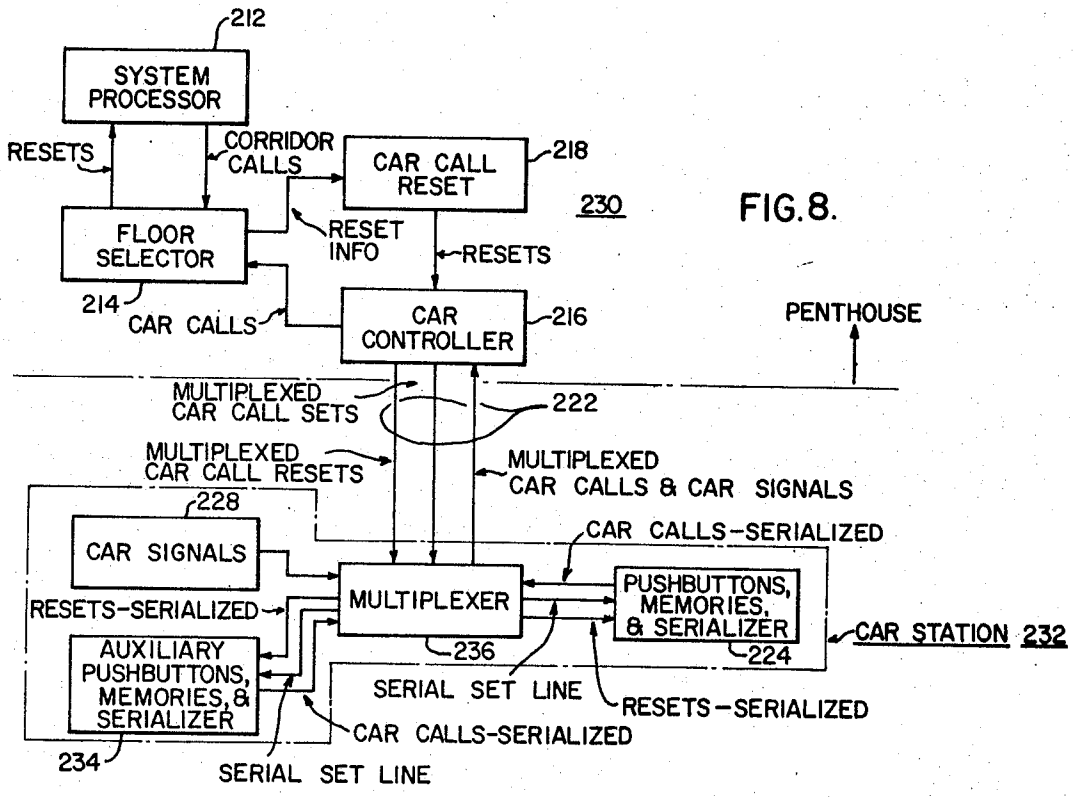

212
SYSTEM PROCESSOR
RESETS
CORRIDOR CALLS
CAR CALL RESET
218
230
FLOOR SELECTOR
RESET INFO
RESETS
214
CAR CALLS
CAR CONTROLLER
216
PENTHOUSE
MULTIPLEXED CAR CALL SETS
222
MULTIPLEXED CAR CALL RESETS
MULTIPLEXED CAR CALLS & CAR SIGNALS
228
CAR CALLS-SERIALIZED
CAR SIGNALS
MULTIPLEXER
PUSHBUTTONS, MEMORIES, & SERIALIZER
CAR STATION 232
RESETS-SERIALIZED
AUXILIARY PUSHBUTTONS, MEMORIES, & SERIALIZER
236
224
SERIAL SET LINE
RESETS-SERIALIZED
CAR CALLS-SERIALIZED
234
SERIAL SET LINE

FIG.8.

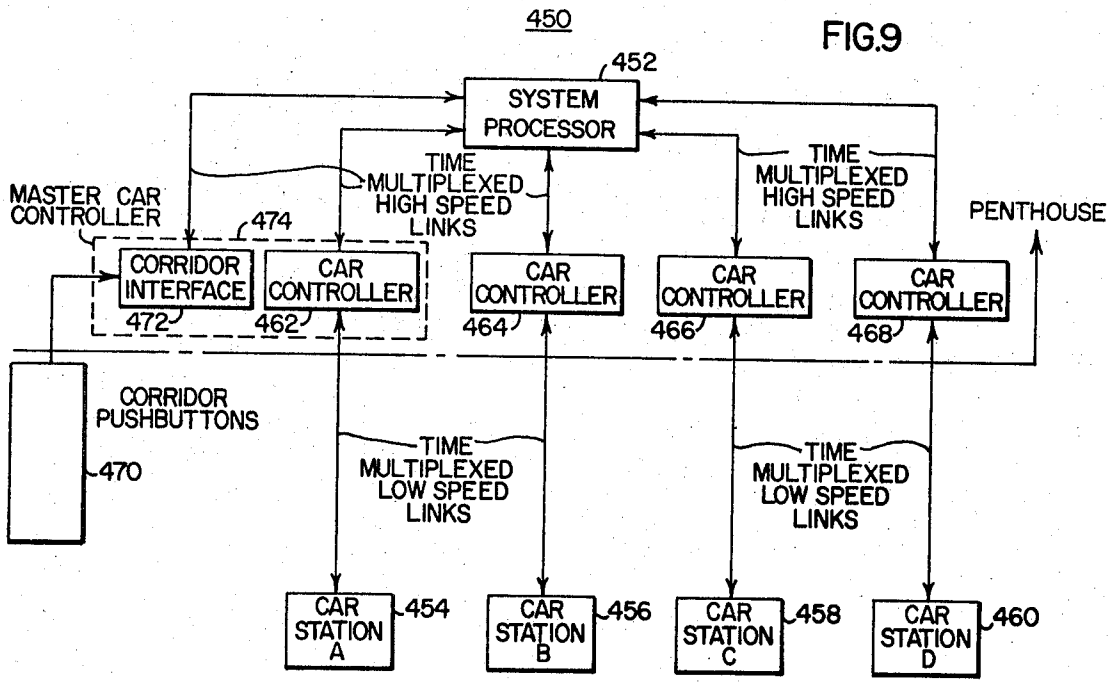

450
452
SYSTEM PROCESSOR
TIME MULTIPLEXED HIGH SPEED LINKS
TIME MULTIPLEXED HIGH SPEED LINKS
MASTER CAR CONTROLLER
474
PENTHOUSE
CORRIDOR INTERFACE
CAR CONTROLLER
CAR CONTROLLER
CAR CONTROLLER
CAR CONTROLLER
472
462
464
466
468
CORRIDOR PUSHBUTTONS
470
TIME MULTIPLEXED LOW SPEED LINKS
TIME MULTIPLEXED LOW SPEED LINKS
CAR STATION A
454
CAR STATION B
456
CAR STATION C
458
CAR STATION D
460

FIG.9

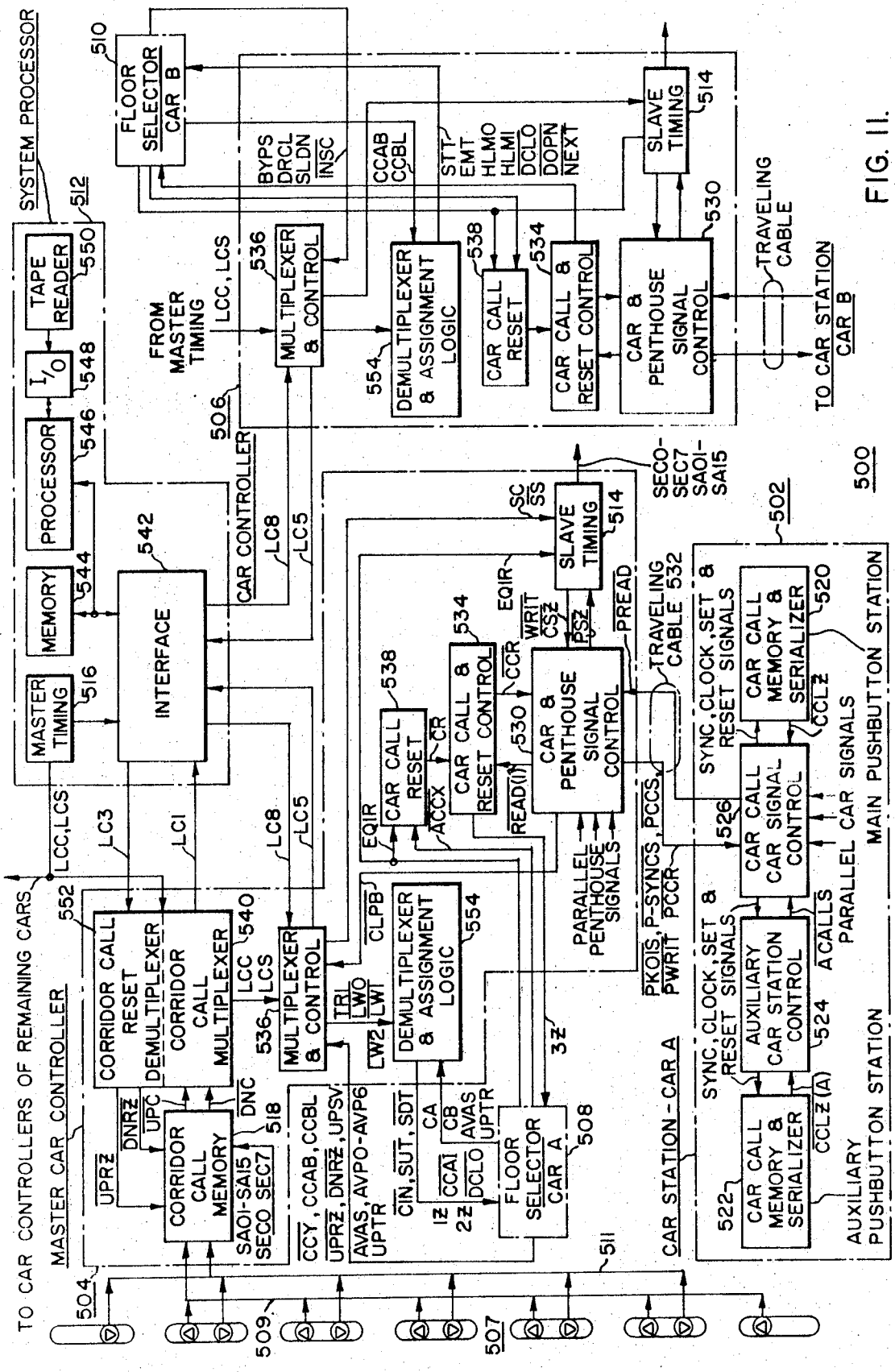

SYSTEM PROCESSOR
TAPE READER
I/O
PROCESSOR
MEMORY
MASTER TIMING
INTERFACE
FLOOR SELECTOR CAR B
FROM MASTER TIMING
MULTIPLEXER & CONTROL
DEMULTIPLEXER & ASSIGNMENT LOGIC
CAR CALL RESET
CAR CALL & RESET CONTROL
CAR & PENTHOUSE SIGNAL CONTROL
SLAVE TIMING
TRAVELING CABLE
TO CAR STATION CAR B
CAR CONTROLLER
TO CAR CONTROLLERS OF REMAINING CARS
MASTER CAR CONTROLLER
CORRIDOR CALL RESET DEMULTIPLEXER
CORRIDOR CALL MULTIPLEXER
CORRIDOR CALL MEMORY
FLOOR SELECTOR CAR A
CAR STATION - CAR A
AUXILIARY CAR STATION CONTROL
CAR CALL MEMORY & SERIALIZER
CAR CALL CAR SIGNAL CONTROL
PARALLEL CAR SIGNALS
MAIN PUSHBUTTON STATION
AUXILIARY PUSHBUTTON STATION
PARALLEL PENTHOUSE SIGNALS
SYNC, CLOCK, SET & RESET SIGNALS
TRAVELING CABLE 532
500
502
504
507
509
511
510
512
514
516
518
520
522
524
526
530
532
534
536
538
540
542
544
546
548
550
552
554

FIG. 11.

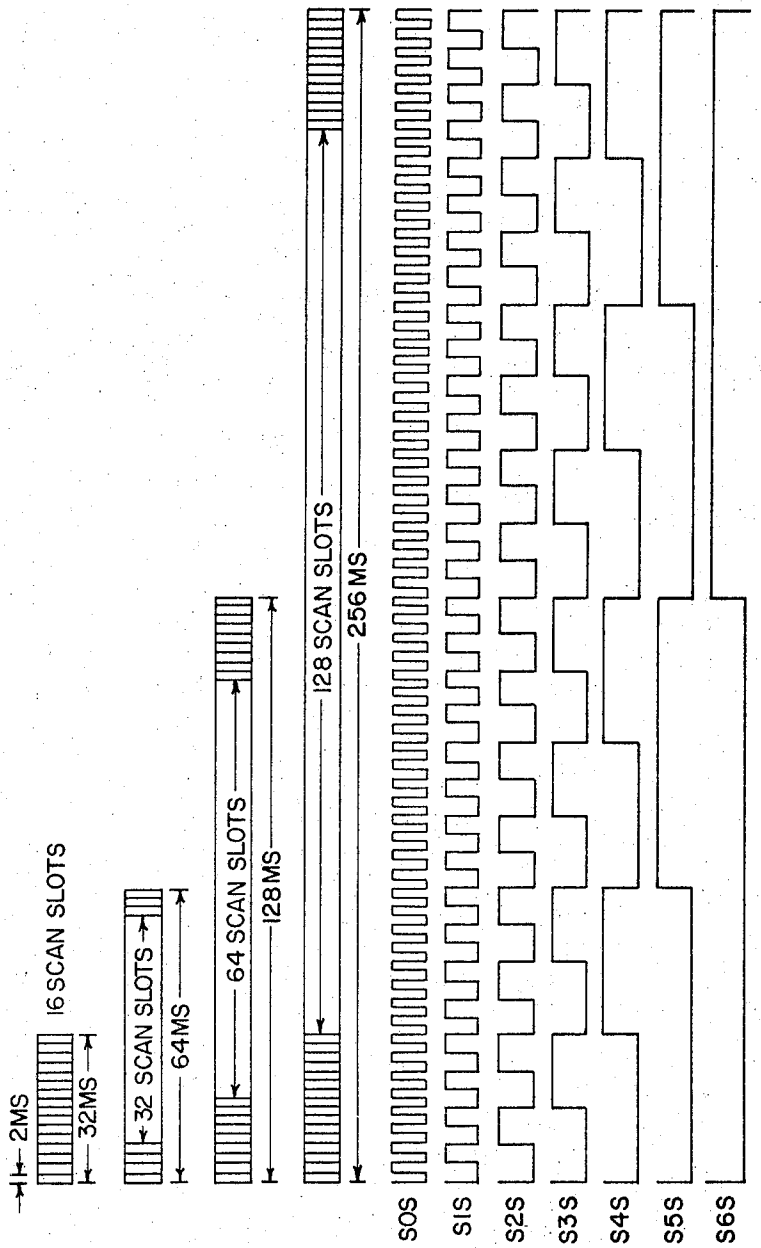

2MS
16 SCAN SLOTS
32MS
32 SCAN SLOTS
64MS
64 SCAN SLOTS
128MS
128 SCAN SLOTS
256 MS
S0S
S1S
S2S
S3S
S4S
S5S
S6S

FIG.12A

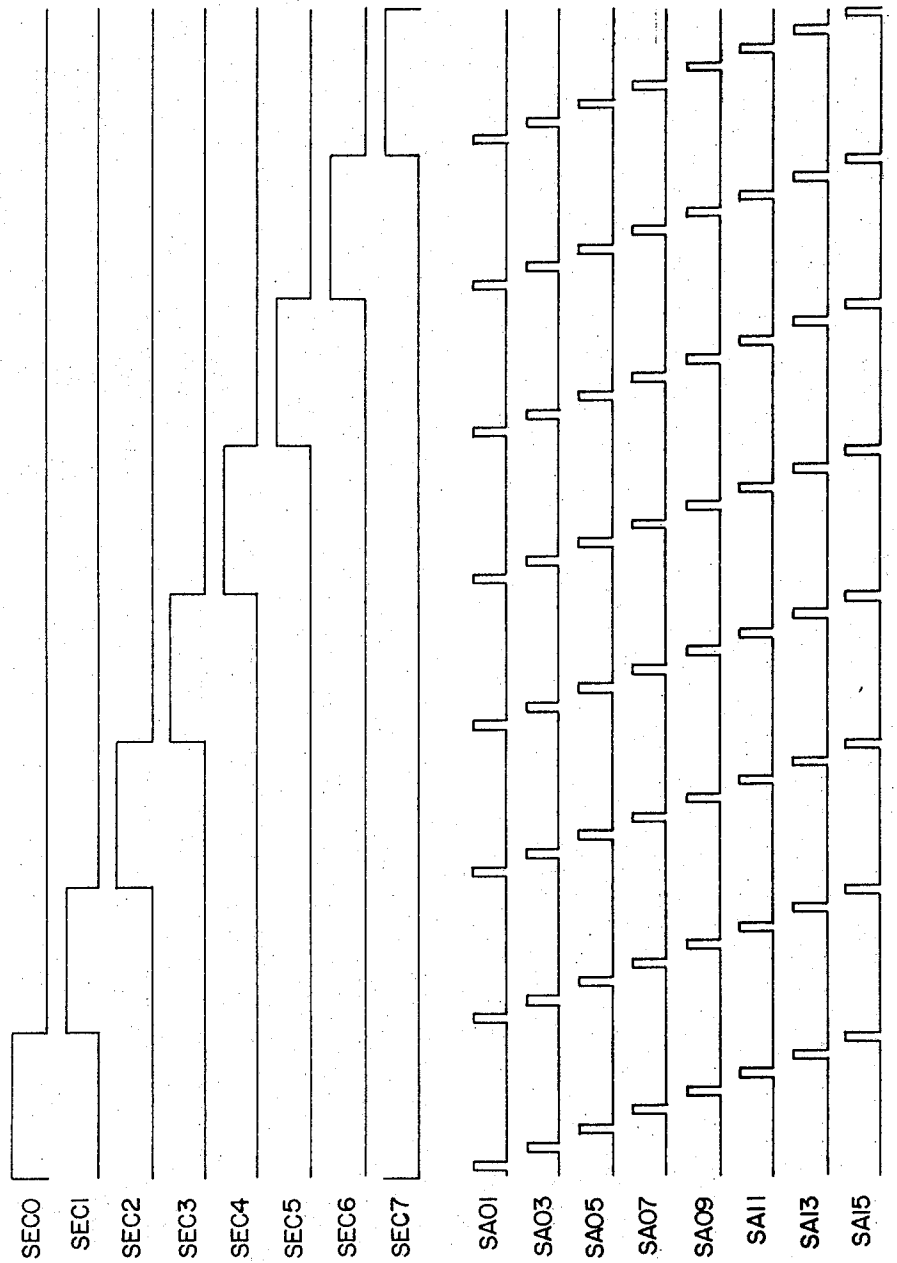

SEC0
SEC1
SEC2
SEC3
SEC4
SEC5
SEC6
SEC7
SA01
SA03
SA05
SA07
SA09
SA11
SA13
SA15

FIG.12B

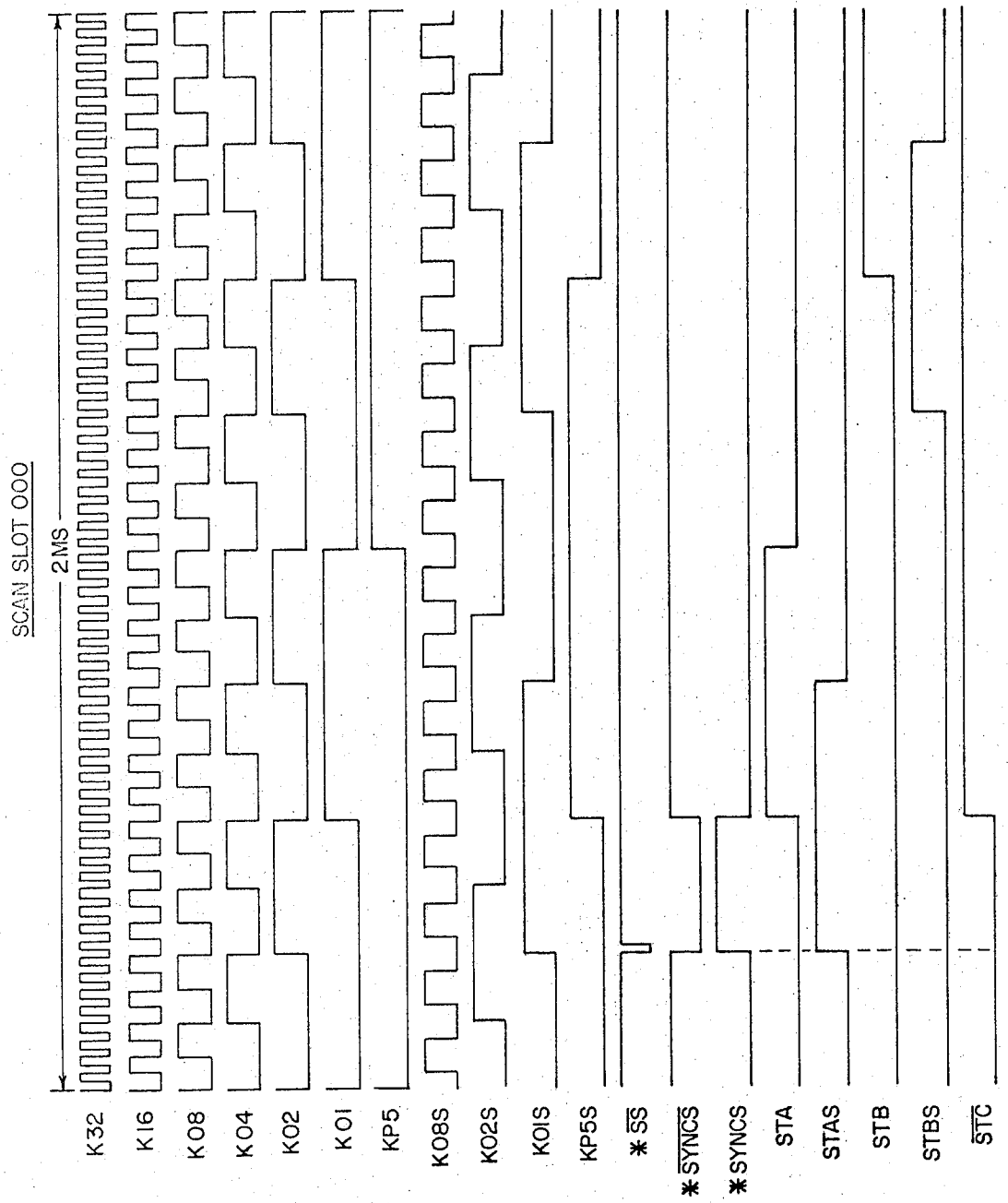

SCAN SLOT 000
2MS
K32
K16
K08
K04
K02
K01
KP5
K08S
K02S
K01S
KP5S
*SS
*SYNCS
*SYNCS
STA
STAS
STB
STBS
STC

FIG.13A

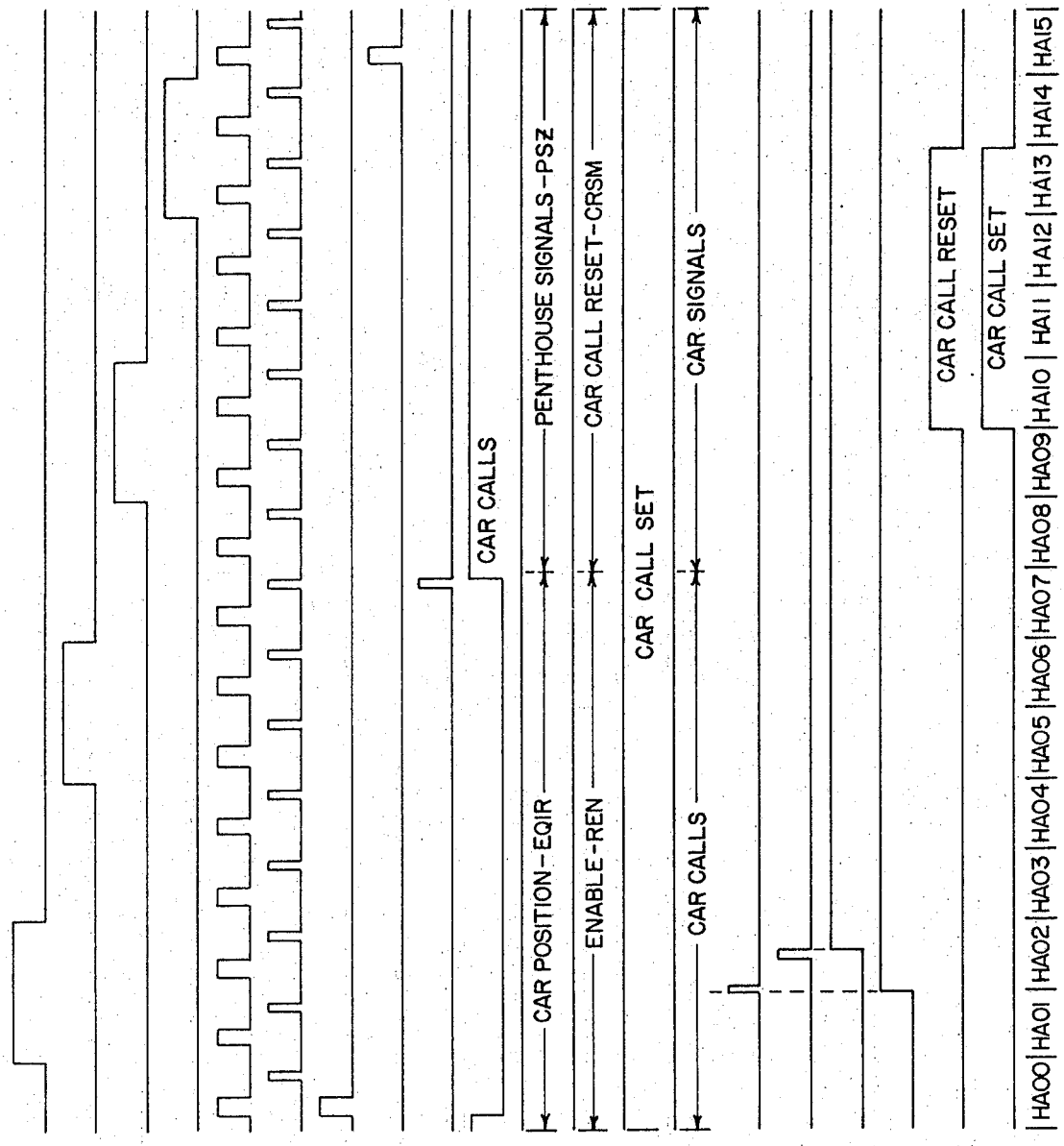

*S100
*S200
*S300
S4
FSTA
FSTB
FSTØ
CSST
CCST
3Z
WRIT
P.WRIT
CCR
PCCR
CCS
PCCS
READ
PREAD
RSYNCO(A)
ASYNC
RCLOCK(A)
ACLK
CCLZ(A)
ACALLS
CRESO(A)
ARES
CSET(A)
ASET
CAR CALLS
CAR POSITION-EQIR
PENTHOUSE SIGNALS-PSZ
ENABLE-REN
CAR CALL RESET-CRSM
CAR CALL SET
CAR CALLS
CAR SIGNALS
CAR CALL RESET
CAR CALL SET
HA00 HA01 HA02 HA03 HA04 HA05 HA06 HA07 HA08 HA09 HA10 HA11 HA12 HA13 HA14 HA15

FIG.13B

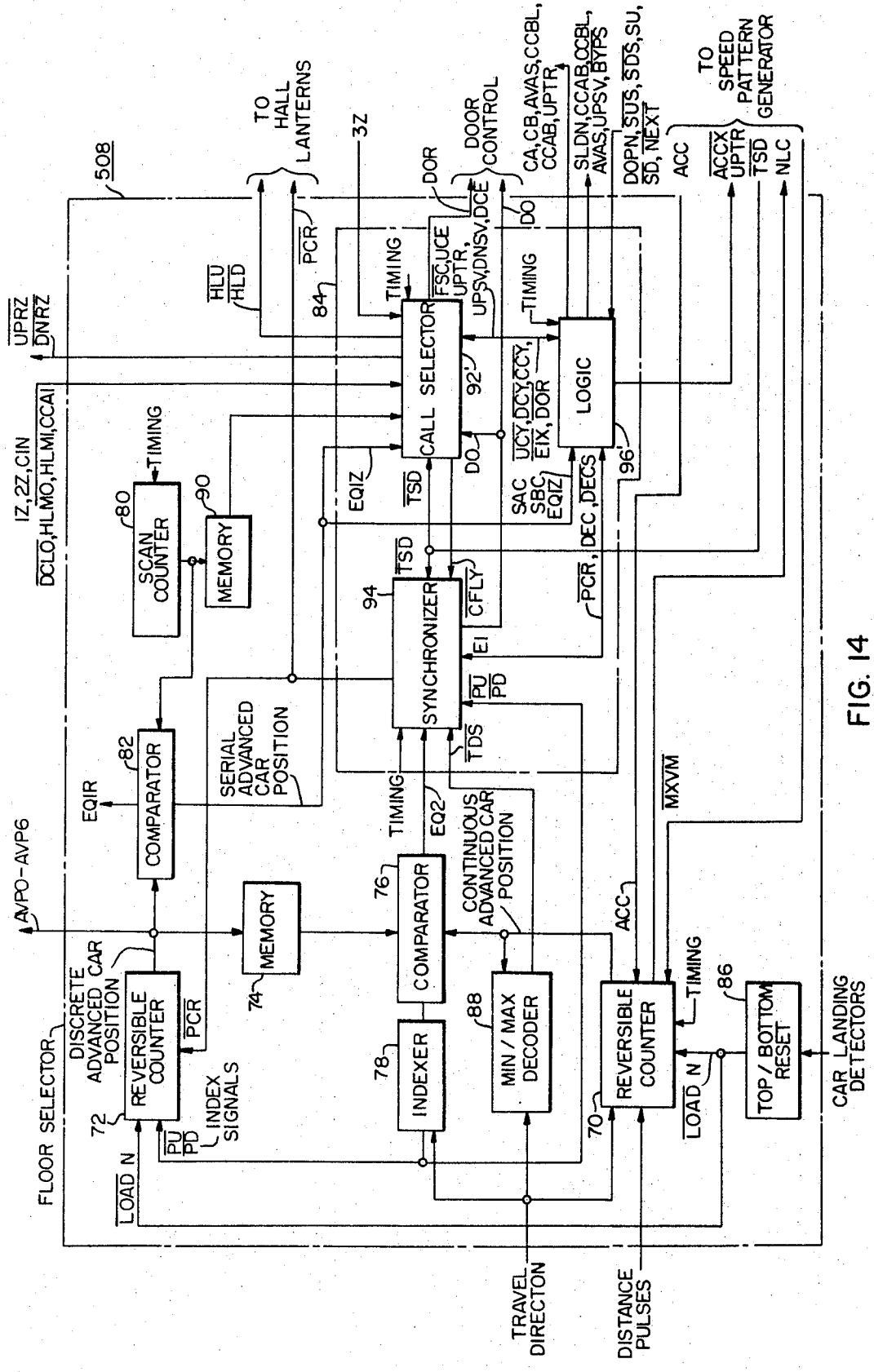

FLOOR SELECTOR
DISCRETE ADVANCED CAR POSITION
REVERSIBLE COUNTER
72
PCR
INDEX SIGNALS
PU
PD
LOAD N
AVP0-AVP6
EQIR
COMPARATOR
82
SERIAL ADVANCED CAR POSITION
SCAN COUNTER
80
TIMING
MEMORY
90
74
MEMORY
76
COMPARATOR
TIMING
EQ2
CONTINUOUS ADVANCED CAR POSITION
INDEXER
78
MIN / MAX DECODER
88
TRAVEL DIRECTION
DISTANCE PULSES
ACC
REVERSIBLE COUNTER
70
MXVM
TIMING
LOAD N
TOP / BOTTOM RESET
86
CAR LANDING DETECTORS
IZ,2Z,CIN
DCLO,HLMO,HLMI,CCAI
UPRZ
DNRZ
508
HLU
HLD
TO HALL LANTERNS
PCR
84
3Z
TIMING
CALL SELECTOR
92
FSC,UCE
UPTR,
UPSV,DNSV,DCE
DOR
DOOR CONTROL
DO
TIMING
LOGIC
96
CA,CB,AVAS,CCBL,
CCAB,UPTR,
SLDN,CCAB,CCBL,
AVAS,UPSV,BYPS
DOPN,SUS,SDS,SU,
SD, NEXT
ACC
TO SPEED PATTERN GENERATOR
ACCX
UPTR
TSD
NLC
EQIZ
TSD
DO
SAC
SBC
EQIZ
UCY,DCY,CCY,
EIX, DOR
DEC,DECS
PCR,
94
TSD
SYNCHRONIZER
CFLY
EI
PU
PD
TDS

FIG. 14

FIG. 15

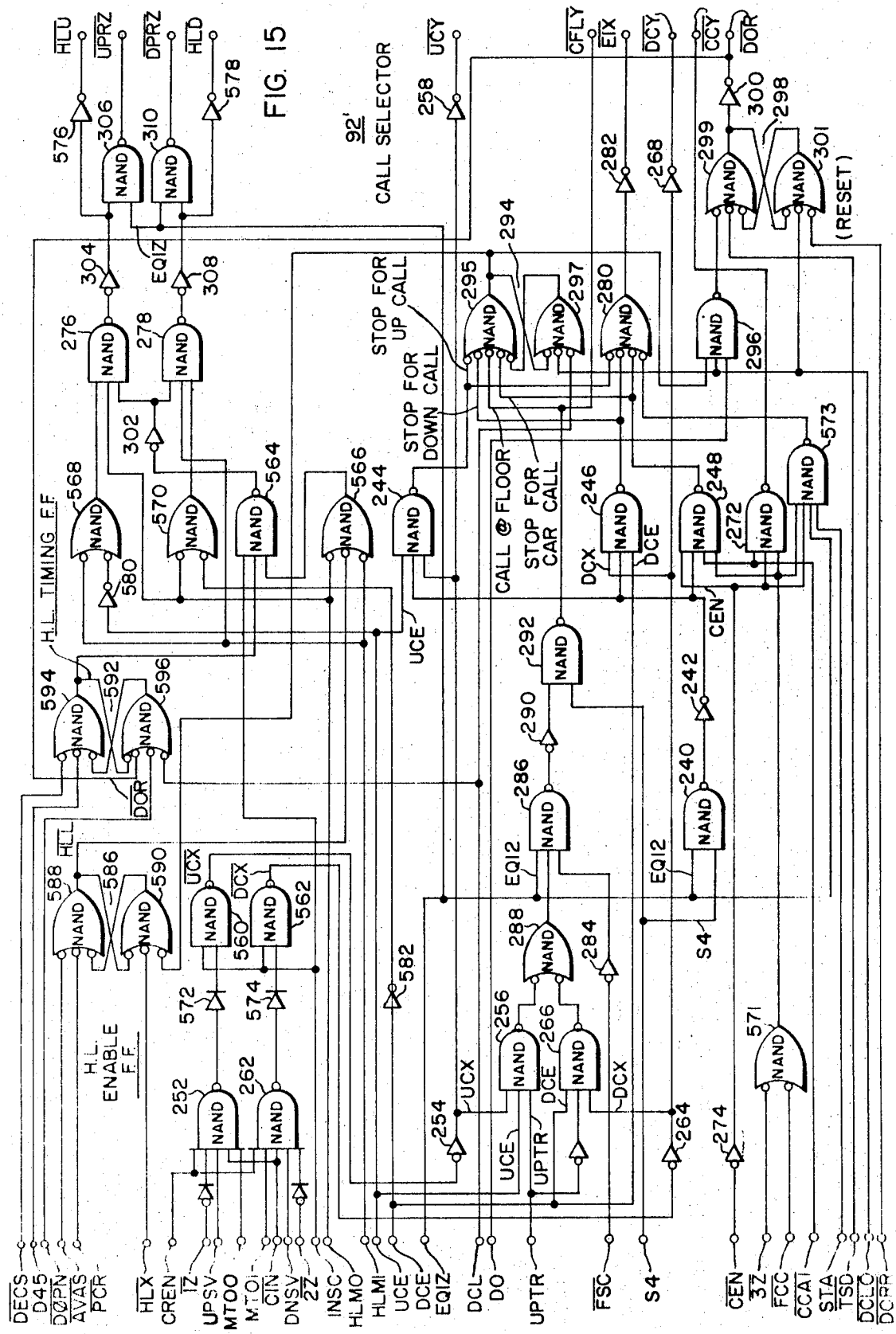

92'
CALL SELECTOR
H.L. TIMING F.F
H.L. ENABLE F.F
STOP FOR UP CALL
STOP FOR DOWN CALL
CALL @ FLOOR
STOP FOR CAR CALL
(RESET)
NAND
HLU
UPRZ
DPRZ
HLD
UCY
CFLY
ETX
DCY
CCY
DOR
EQIZ
UCE
DCE
DCX
CEN
EQ12
UCX
DCX
UPTR
HLL
HLX
DECS
D45
DOPN
AVAS
PCR
CREN
IZ
UPSV
MTOO
MTOI
CIN
DNSV
2Z
INSC
HLMO
HLMI
DCL
DO
FSC
S4
3Z
FCC
CCAI
STA
TSD
DCLO
DCRR

FIG. 16A.

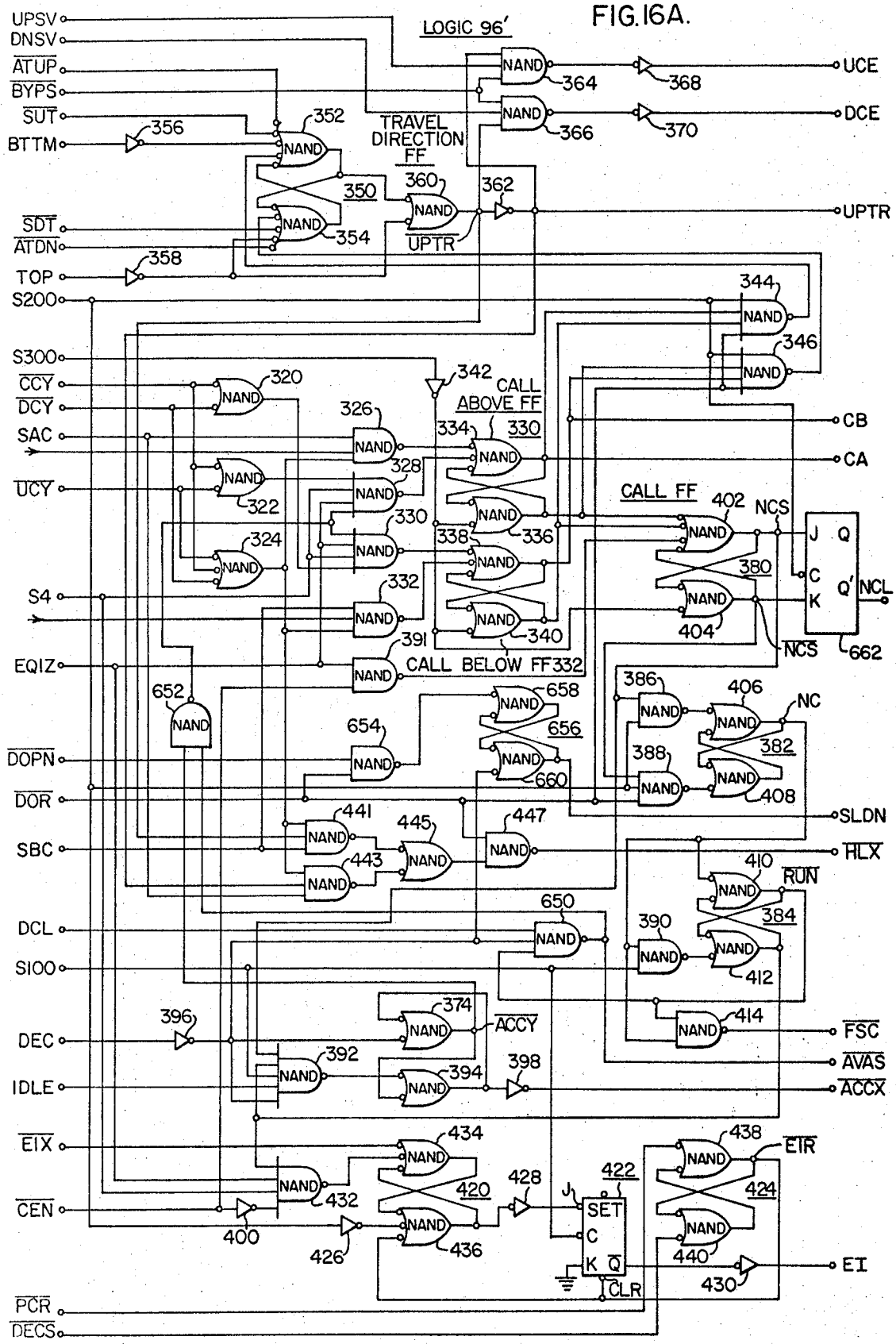
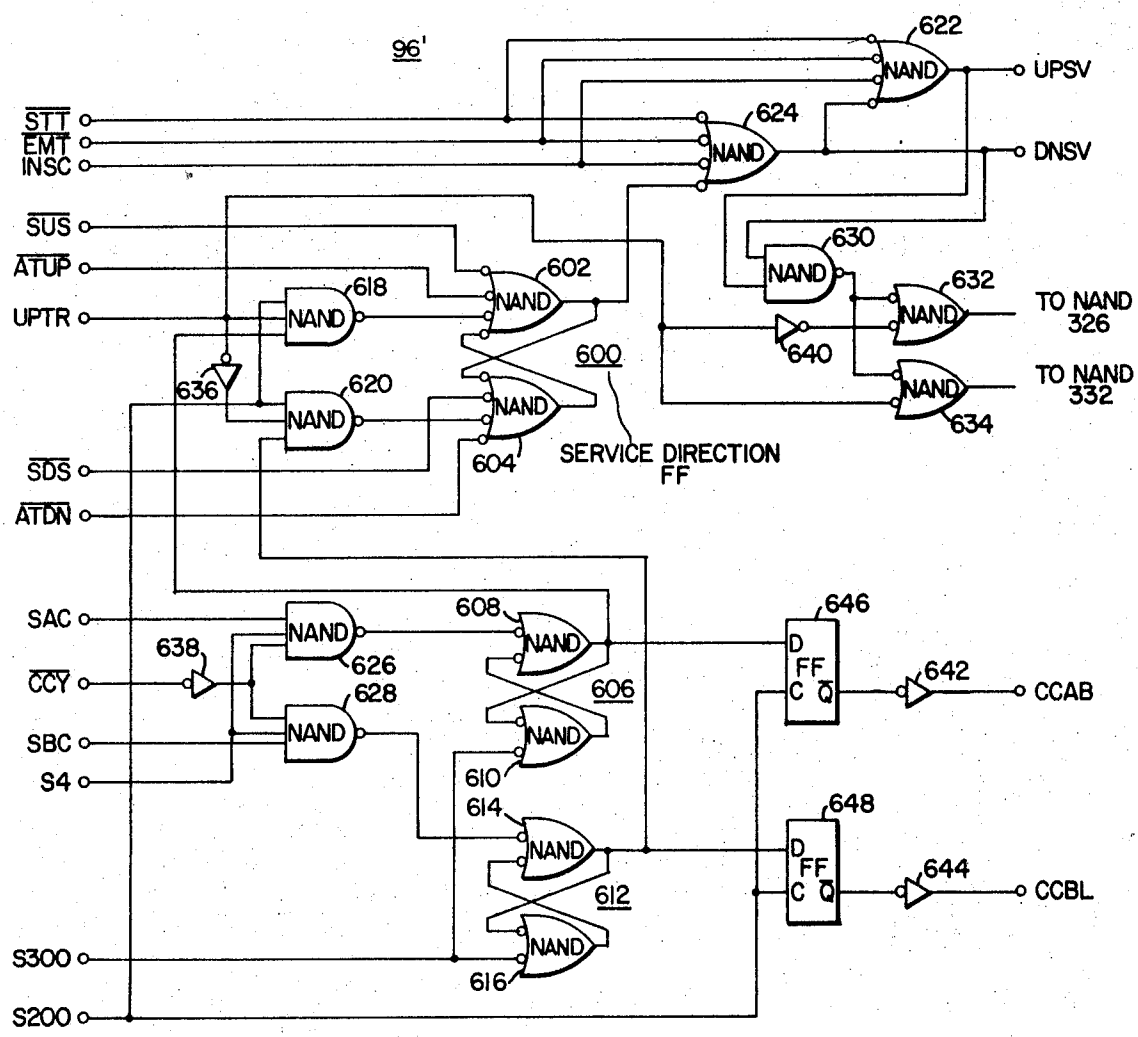
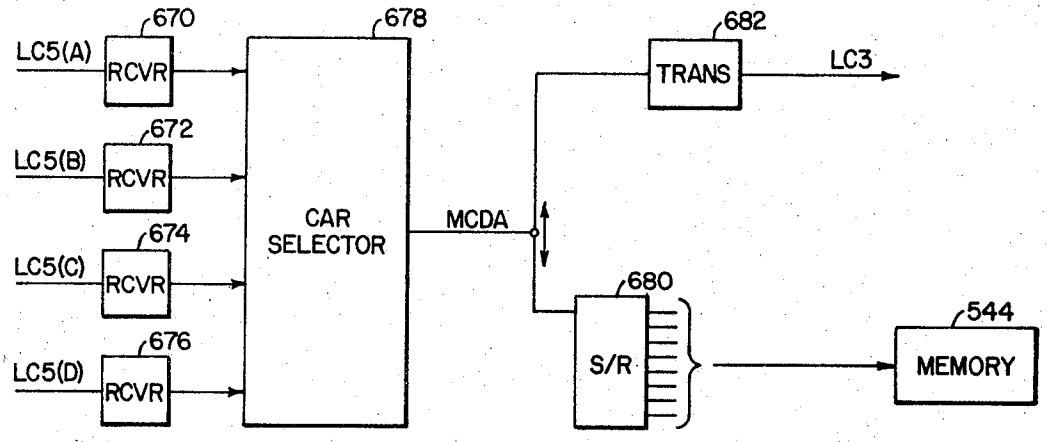

UPSV
DNSV
ATUP
BYPS
SUT
BTTM
SDT
ATDN
TOP
S200
S300
CCY
DCY
SAC
UCY
S4
EQIZ
DOPN
DOR
SBC
DCL
SIOO
DEC
IDLE
EIX
CEN
PCR
DECS
LOGIC 96'
NAND
364
368
UCE
366
370
DCE
352
356
TRAVEL DIRECTION FF
350
354
360
362
UPTR
358
344
346
320
342
CALL ABOVE FF
326
334
330
CB
CA
322
328
324
CALL FF
402
NCS
338
336
380
404
332
340
391
CALL BELOW FF332
662
652
658
656
654
660
386
406
NC
382
388
408
SLDN
441
445
447
443
HLX
410
RUN
650
384
390
412
374
414
FSC
ACCY
AVAS
392
394
398
ACCX
396
400
434
438
EIR
420
428
422
424
J
SET
C
K
Q
CLR
432
426
436
440
430
EI

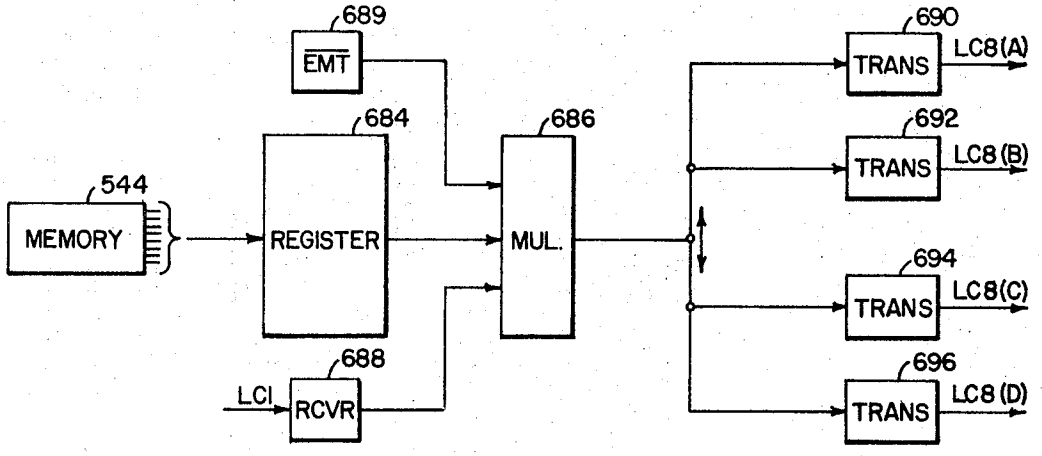
FIG.16B


96'
622
NAND
UPSV
624
NAND
DNSV
STT
EMT
INSC
SUS
ATUP
UPTR
618
NAND
602
NAND
630
NAND
632
NAND
TO NAND 326
640
636
620
NAND
600
NAND
NAND
TO NAND 332
634
604
SERVICE DIRECTION FF
SDS
ATDN
SAC
CCY
SBC
S4
638
NAND
626
628
NAND
608
NAND
606
NAND
610
646
D FF C Q
642
CCAB
614
NAND
612
NAND
616
648
D FF C Q
644
CCBL
S300
S200



INPUT CHANNEL 669
542

FIG.17



OUTPUT CHANNEL 683
542

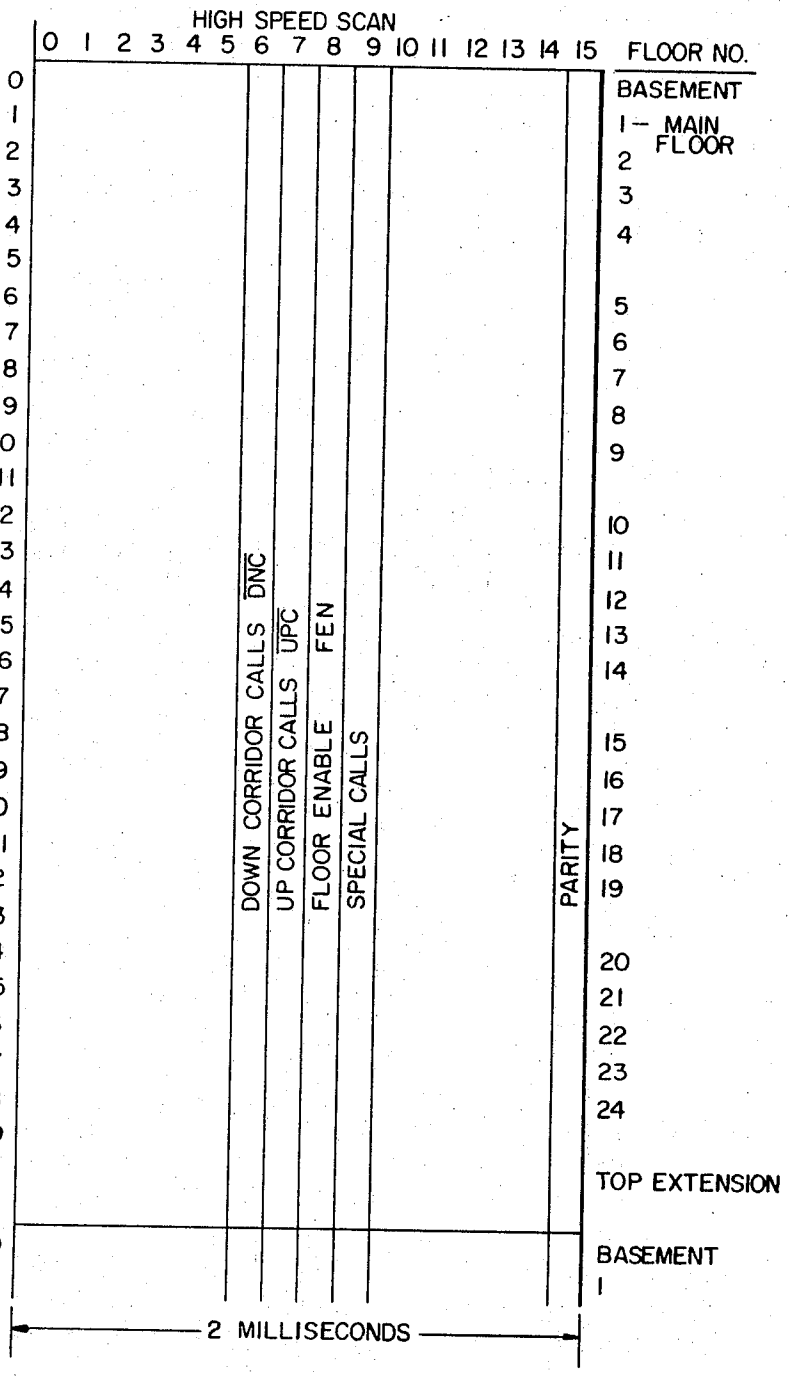
FIG.18

FIG. 19

DATA LINK LCI

KO4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
KO2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
KO1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
KP5	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

S6S	S5S	S4S	S3S	S2S	S1S	S0S	SCAN SLOTS (2 MS EACH)
0	0	0	0	0	0	0	0
		0	0	0	0	1	1
		0	0	0	1	0	2
		0	0	0	1	1	3
		0	0	1	0	0	4
		0	0	1	0	1	5
		0	0	1	1	0	6
		0	0	1	1	1	7
		0	1	0	0	0	8
		0	1	0	0	1	9
		0	1	0	1	0	10
		0	1	0	1	1	11
		0	1	1	0	0	12
		0	1	1	0	1	13
		0	1	1	1	0	14
		0	1	1	1	1	15
		1	0	0	0	0	16
		1	0	0	0	1	17
		1	0	0	1	0	18
		1	0	0	1	1	19
		1	0	1	0	0	20
		1	0	1	0	1	21
		1	0	1	1	0	22
		1	0	1	1	1	23
		1	1	0	0	0	24
		1	1	0	0	1	25
		1	1	0	1	0	26
		1	1	0	1	1	27
		1	1	1	0	0	28
		1	1	1	0	1	29
		1	1	1	1	0	30
		1	1	1	1	1	31
0	0	0	0	0	0	0	0
		0	0	0	0	1	1



DATA LINK LC3

FIG. 20

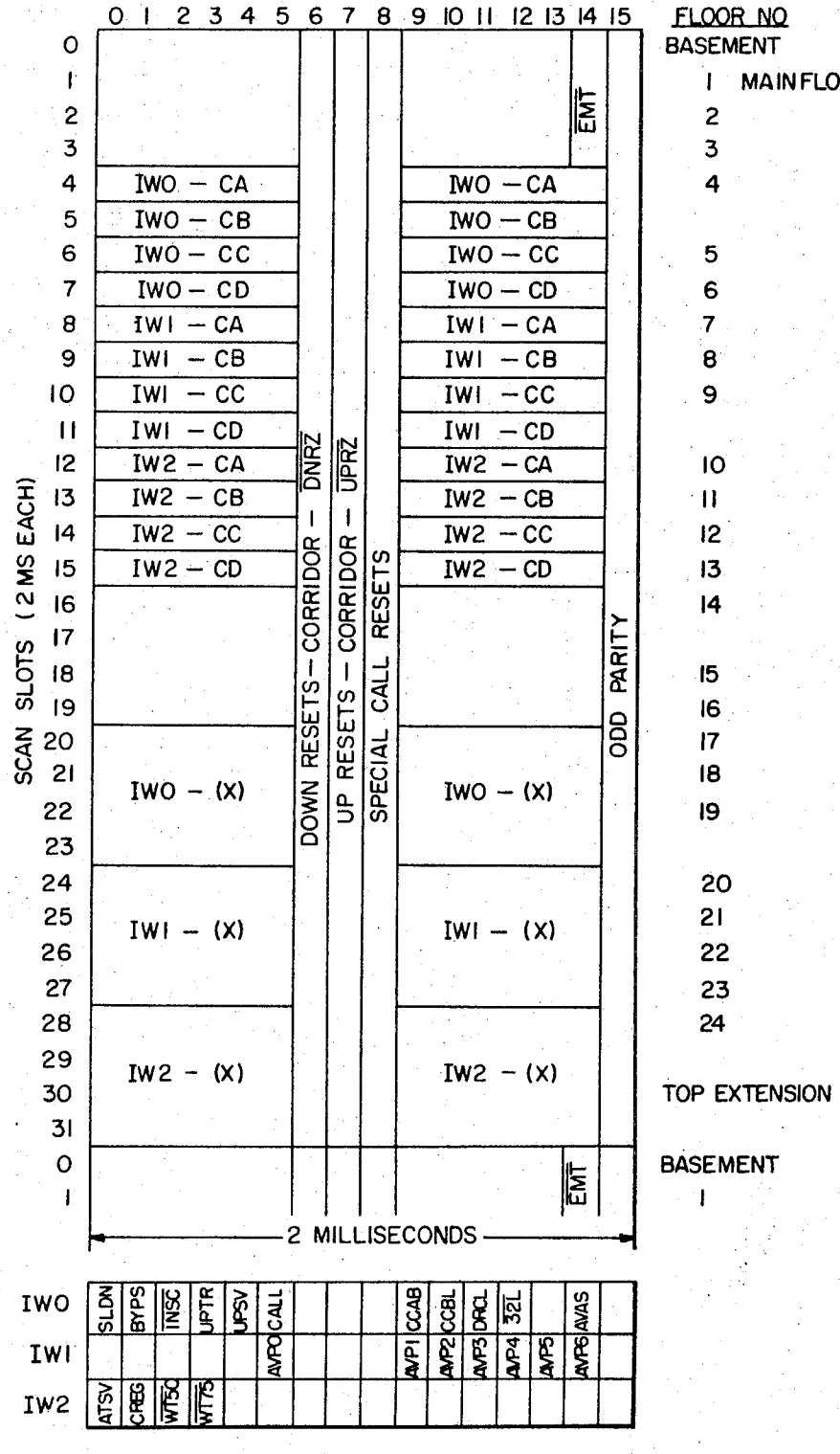


FIG. 21

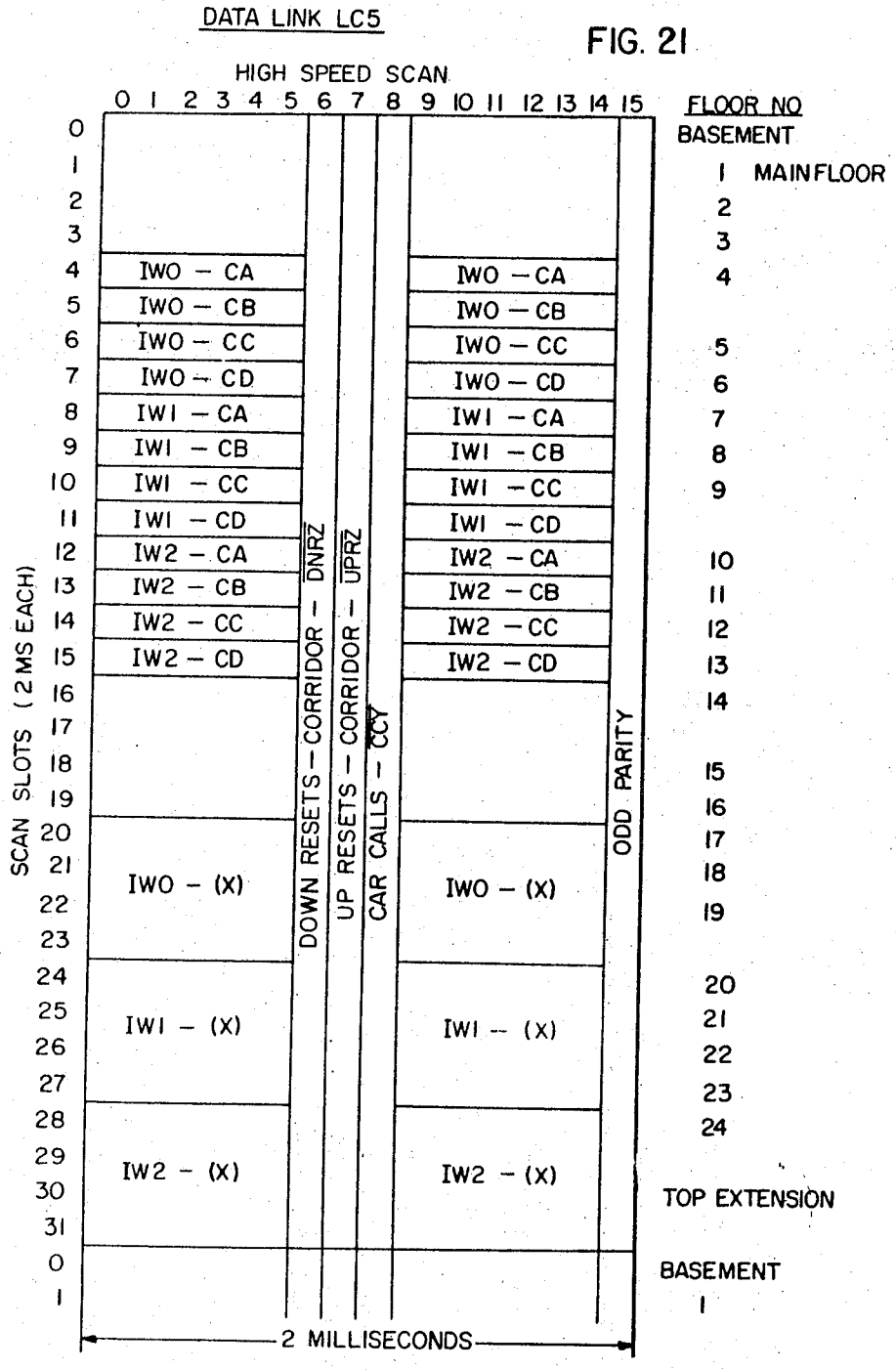
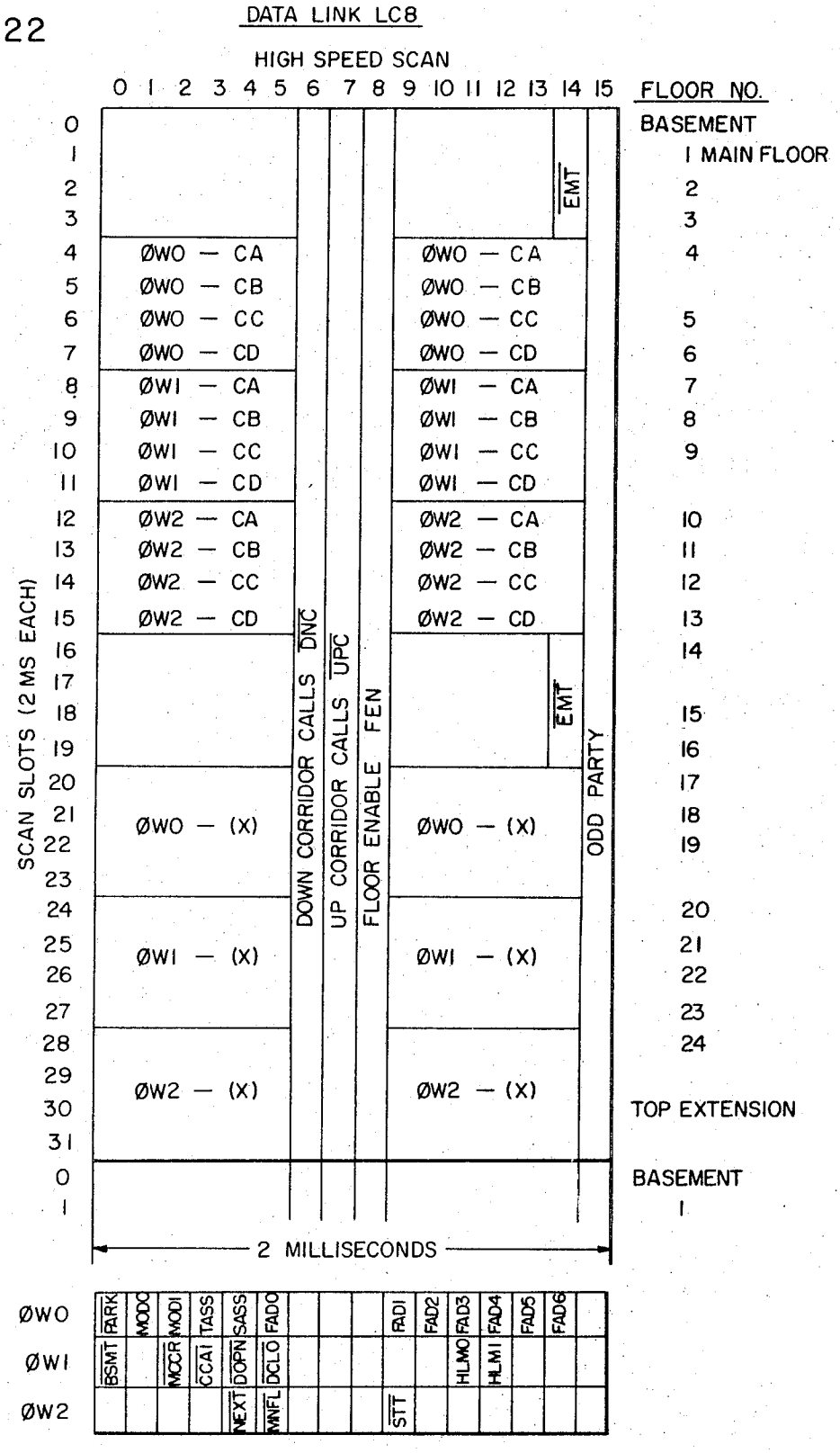

DATA LINK LC5
HIGH SPEED SCAN
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
SCAN SLOTS (2 MS EACH)
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 0 1
IWO – CA
IWO – CB
IWO – CC
IWO – CD
IW1 – CA
IW1 – CB
IW1 – CC
IW1 – CD
IW2 – CA
IW2 – CB
IW2 – CC
IW2 – CD
IWO – (X)
IW1 – (X)
IW2 – (X)
DOWN RESETS – CORRIDOR – DNRZ
UP RESETS – CORRIDOR – UPRZ
CAR CALLS – CCY
ODD PARITY
2 MILLISECONDS
FLOOR NO
BASEMENT
1 MAIN FLOOR
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
TOP EXTENSION
BASEMENT
1

FIG. 22


DATA LINK LC8
HIGH SPEED SCAN
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
FLOOR NO.
SCAN SLOTS (2 MS EACH)
0 BASEMENT
1 MAIN FLOOR
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
0
1
TOP EXTENSION
BASEMENT
ØWO — CA
ØWO — CB
ØWO — CC
ØWO — CD
ØWI — CA
ØWI — CB
ØWI — CC
ØWI — CD
ØW2 — CA
ØW2 — CB
ØW2 — CC
ØW2 — CD
ØWO — (X)
ØWI — (X)
ØW2 — (X)
EMT
EMT
DOWN CORRIDOR CALLS DNC
UP CORRIDOR CALLS UPC
FLOOR ENABLE FEN
ODD PARITY
2 MILLISECONDS
ØWO
ØWI
ØW2
PARK
MODO
MODI
TASS
SASS
FADO
FADI
FAD2
FAD3
FAD4
FAD5
FAD6
BSMT
MCCR
CCAI
DOPN
NEXT
DCLO
MNFL
STT
HLMO
HLMI

CORE MAP

	WORD NAME	ADDRESS	BIT 11	10	9	8	7	6	5	4	3	2	1	BIT 0
CORRIDOR & CAR CALLS	CL 000	0 1111 0000 0000			UC 000	DC 000					CC 000 (CAR D)	CC 000 (CAR C)	CC 000 (CAR B)	CC 000 (CAR A)
	CL 001	0000 0001			UC 001	DC 001					CC 001 (D)	CC 001 (C)	CC 001 (B)	CC 001 (A)
	CL 002	0000 0010			UC 002	DC 002					CC 002 (D)	CC 002 (C)	CC 002 (B)	CC 002 (A)
	CL 125	0111 1101			UC 125	DC 125					CC 125	CC 125	CC 125	CC 125
	CL 126	0111 1110			UC 126	DC 126					CC 126	CC 126	CC 126	CC 126
	CL 127	0111 1111			UC 127	DC 127					CC 127 (D)	CC 127 (C)	CC 127 (B)	CC 127 (A)
CAR SIGNALS CAR A	IWO-CA	0 1111 1001 0000	AVAS		$\overline{32L}$	DRCL	CCBL	CCAB	CALL	UPSV	UPTR	$\overline{INSC}$	BYPS	SLDN
	IW1-CA	1010 0000	AVP6	AVP5	AVP4	AVP3	AVP2	AVP1	AVPO					
	IW2-CA	1011 0000									$\overline{WT75}$	$\overline{WT50}$	CREG	ATSV
	ØWO-CA	1000 1101	FAD6	FAD5	FAD4	FAD3	FAD2	FAD1	FADO	SASS	TASS	MØD1	MØDO	$\overline{PARK}$
	ØW1-CA	1001 1101			HLM1	HLMO			$\overline{DCLØ}$	$\overline{DOPN}$	$\overline{CCAI}$	$\overline{MCCR}$		$\overline{BSMT}$
	ØW2-CA	1010 1101						$\overline{STT}$	$\overline{MNFL}$	$\overline{NEXT}$				
CAR SIGNALS CAR B	IWO-CB	1001 0010												
	IW1-CB	1010 0010												
	IW2-CB	1011 0010				(SAME AS CAR A)								
	ØWO-CB	1000 1111												
	ØW1-CB	1001 1111												
	ØW2-CB	1010 1111												

REMAINING CARS OF BANK

FIG. 23

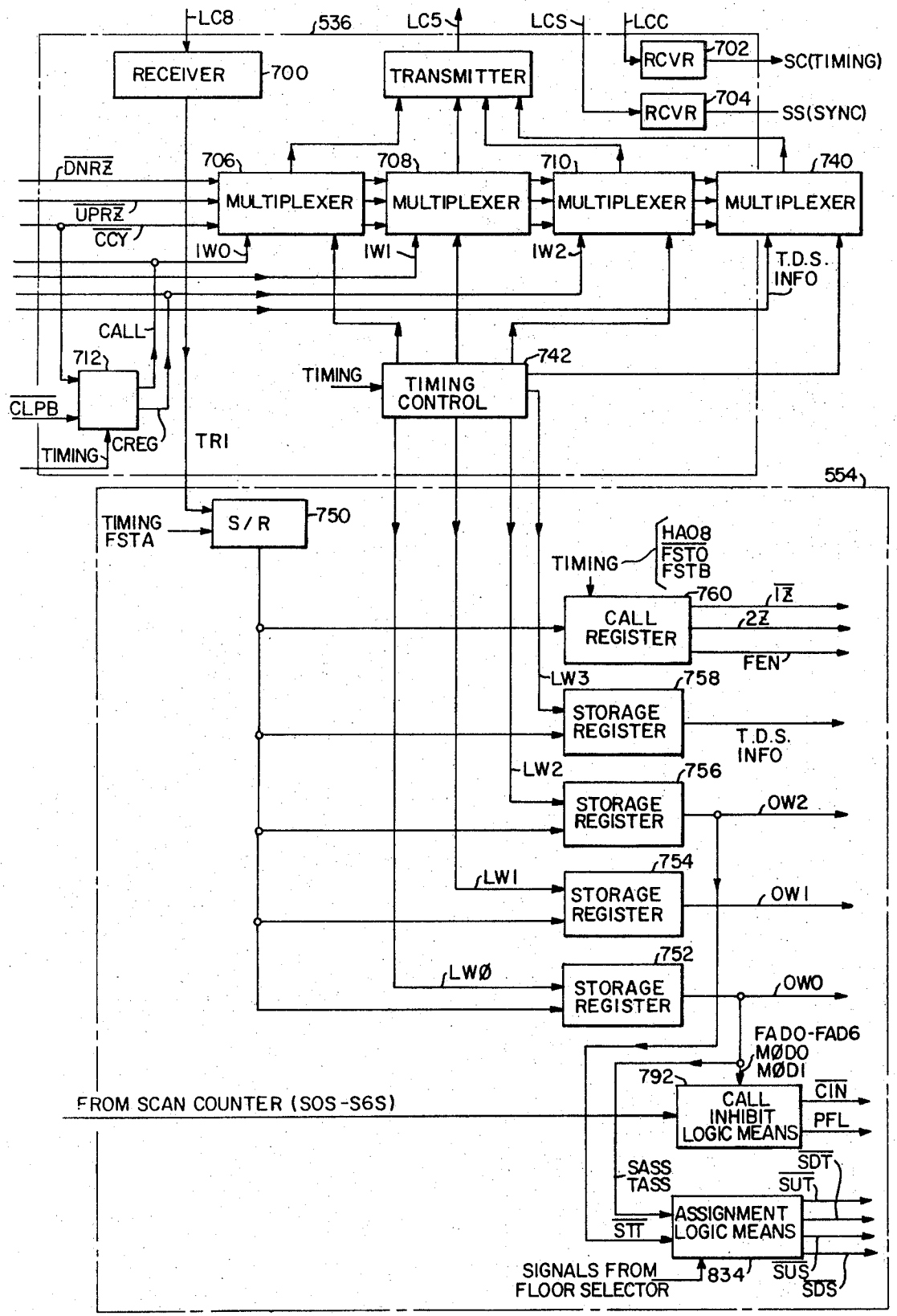

LC8
536
LC5
LCS
LCC
RECEIVER
700
TRANSMITTER
RCVR
702
SC(TIMING)
RCVR
704
SS(SYNC)
DNRZ
UPRZ
CCY
706
708
710
740
MULTIPLEXER
MULTIPLEXER
MULTIPLEXER
MULTIPLEXER
IWO
IWI
IW2
T.D.S. INFO
CALL
712
CLPB
CREG
TRI
TIMING
TIMING CONTROL
742
554
TIMING FSTA
S/R
750
HAO8
FSTO
FSTB
TIMING
CALL REGISTER
760
1Z
2Z
FEN
LW3
758
STORAGE REGISTER
T.D.S. INFO
LW2
756
STORAGE REGISTER
OW2
754
LWI
STORAGE REGISTER
OWI
752
LWØ
STORAGE REGISTER
OWO
FA DO-FAD6
MØDO
MØDI
792
FROM SCAN COUNTER (SOS-S6S)
CALL INHIBIT LOGIC MEANS
CIN
PFL
SASS
TASS
SDT
SUT
STT
ASSIGNMENT LOGIC MEANS
SIGNALS FROM FLOOR SELECTOR
834
SUS
SDS

FIG. 24

FIG.25

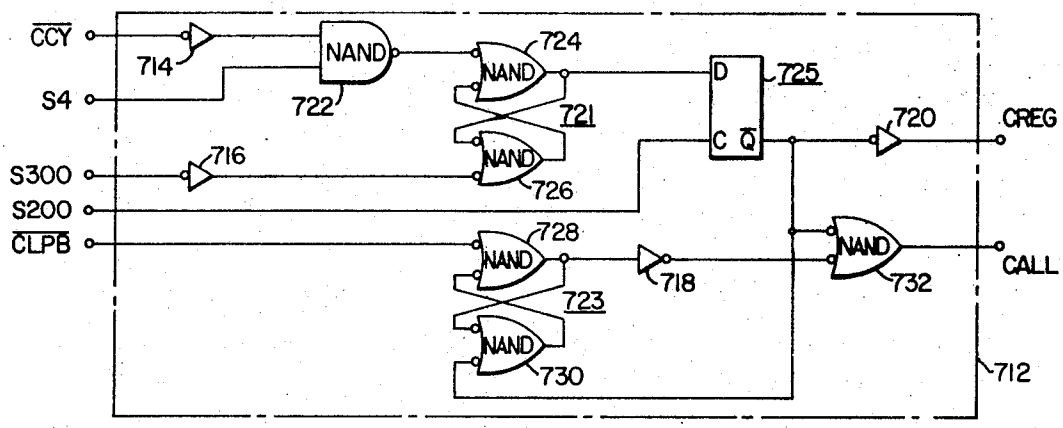


FIG.26

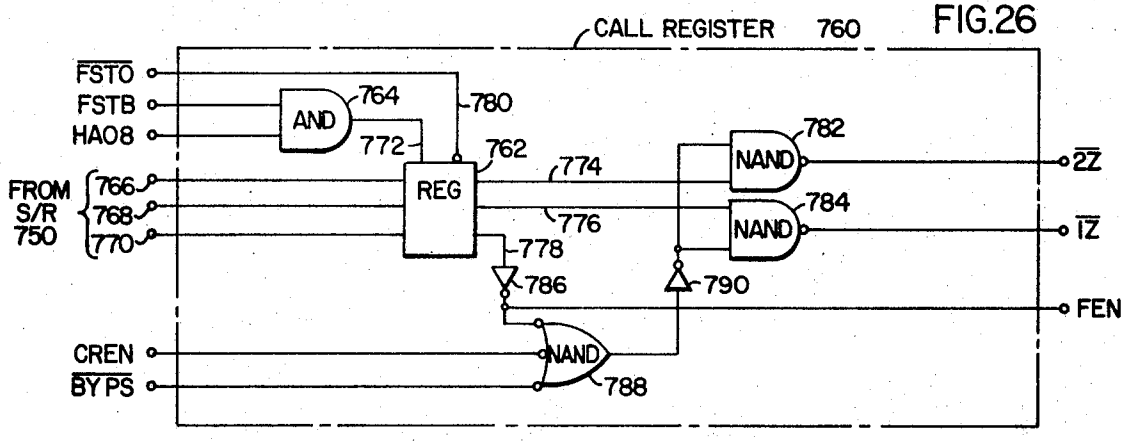


FIG.27

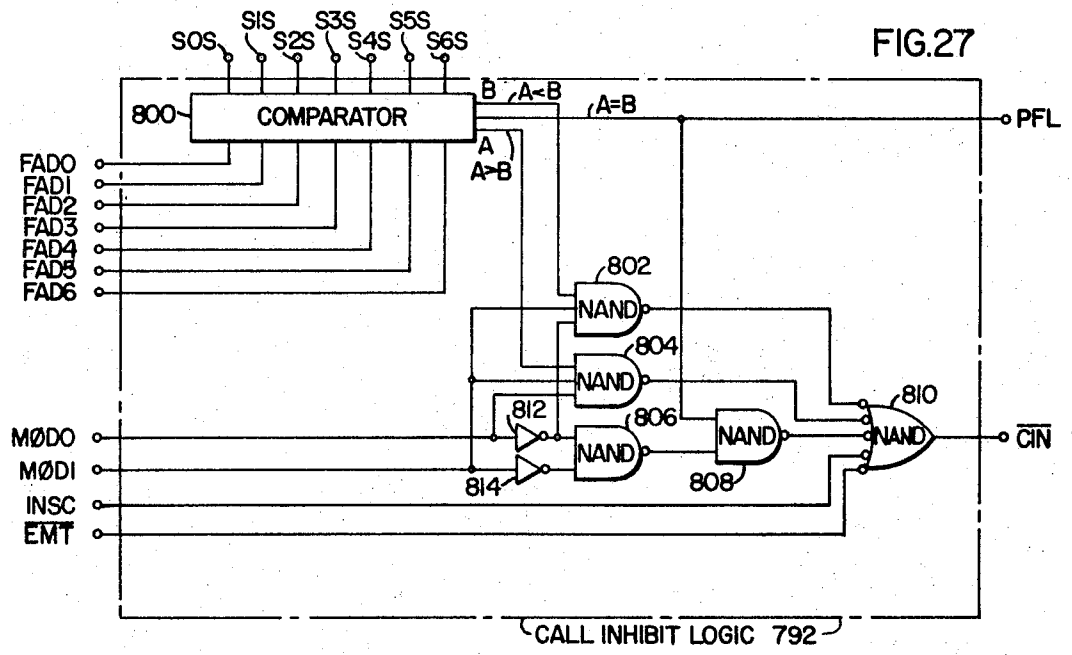
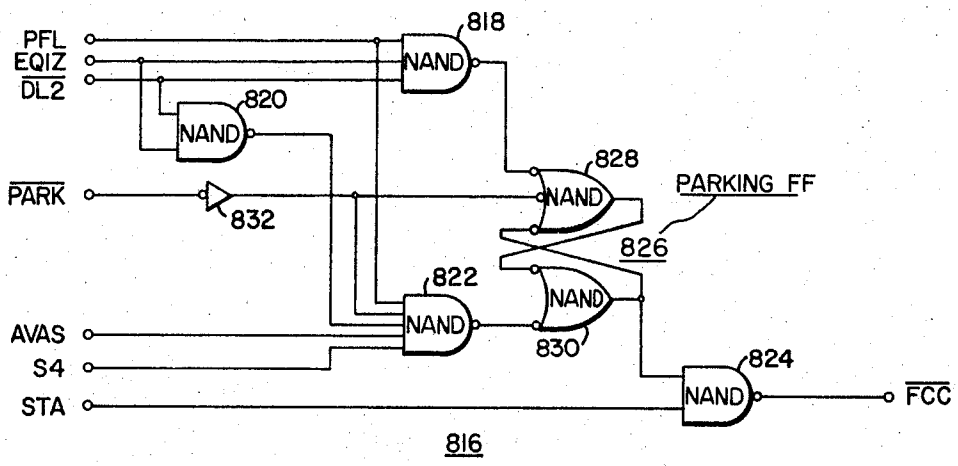


FIG.28


PFL
EQIZ
DL2
PARK
AVAS
S4
STA
NAND
818
820
828
832
822
830
824
826
PARKING FF
FCC
816

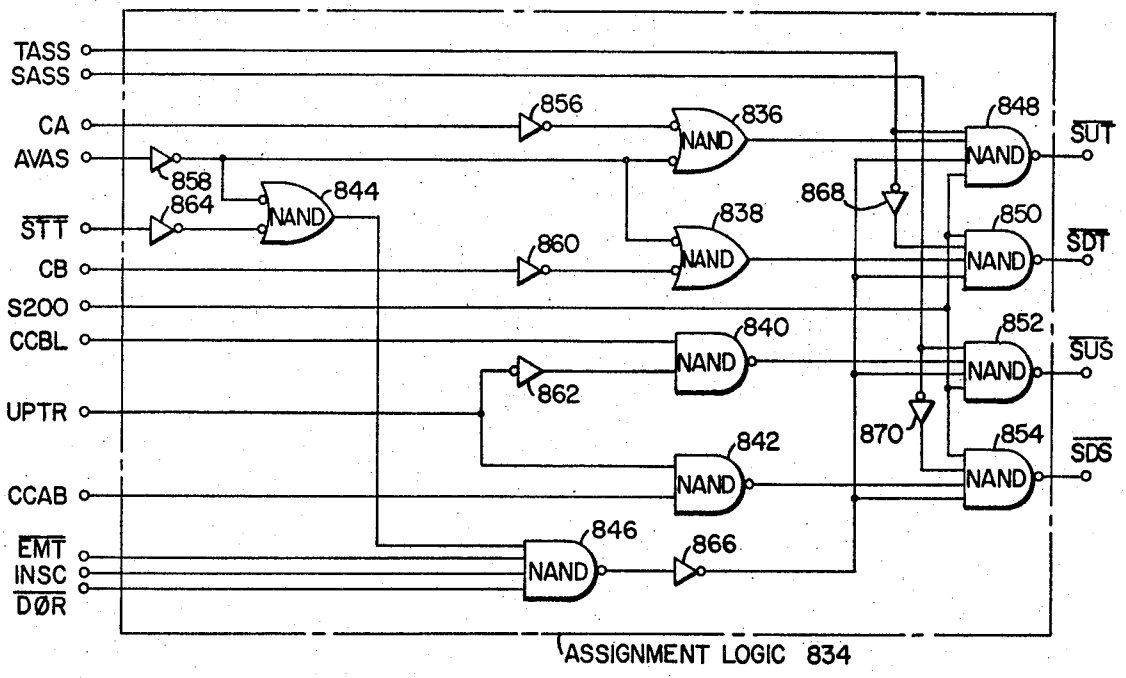

TASS
SASS
CA
AVAS
STT
CB
S200
CCBL
UPTR
CCAB
EMT
INSC
DØR
NAND
856
836
858
844
864
838
860
840
862
842
846
866
848
868
850
852
870
854
SUT
SDT
SUS
SDS
ASSIGNMENT LOGIC 834

FIG.29

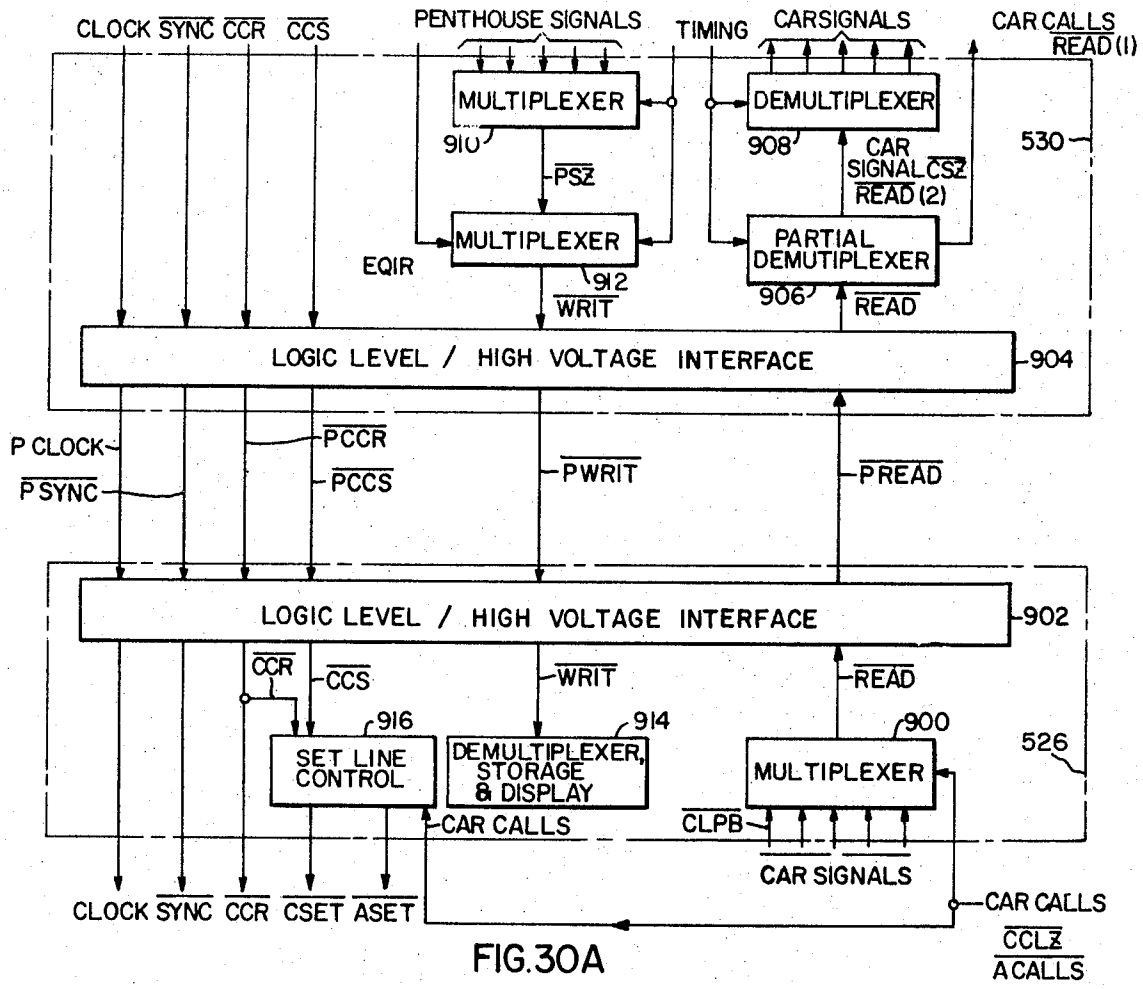


FIG.30A

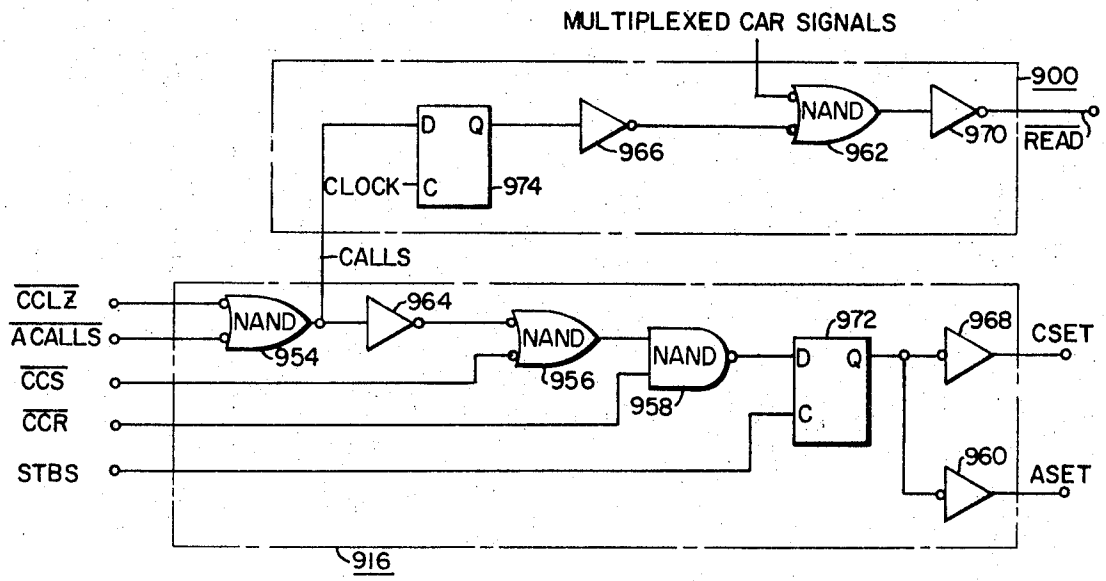


FIG.30B

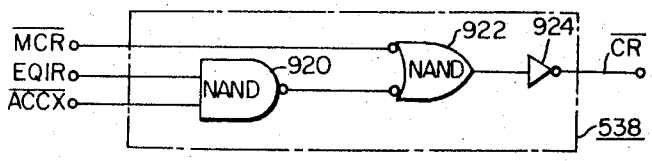


FIG. 31

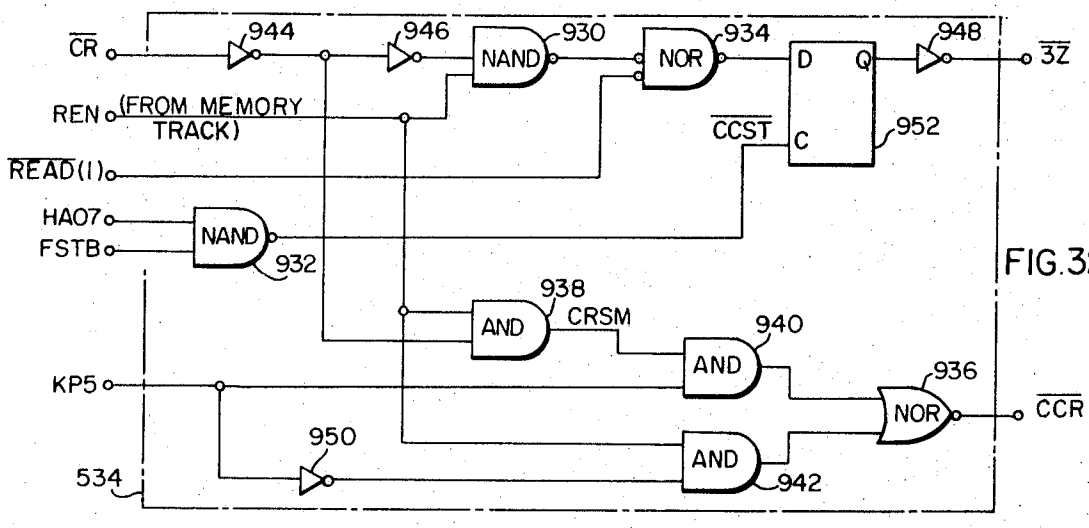


FIG.32

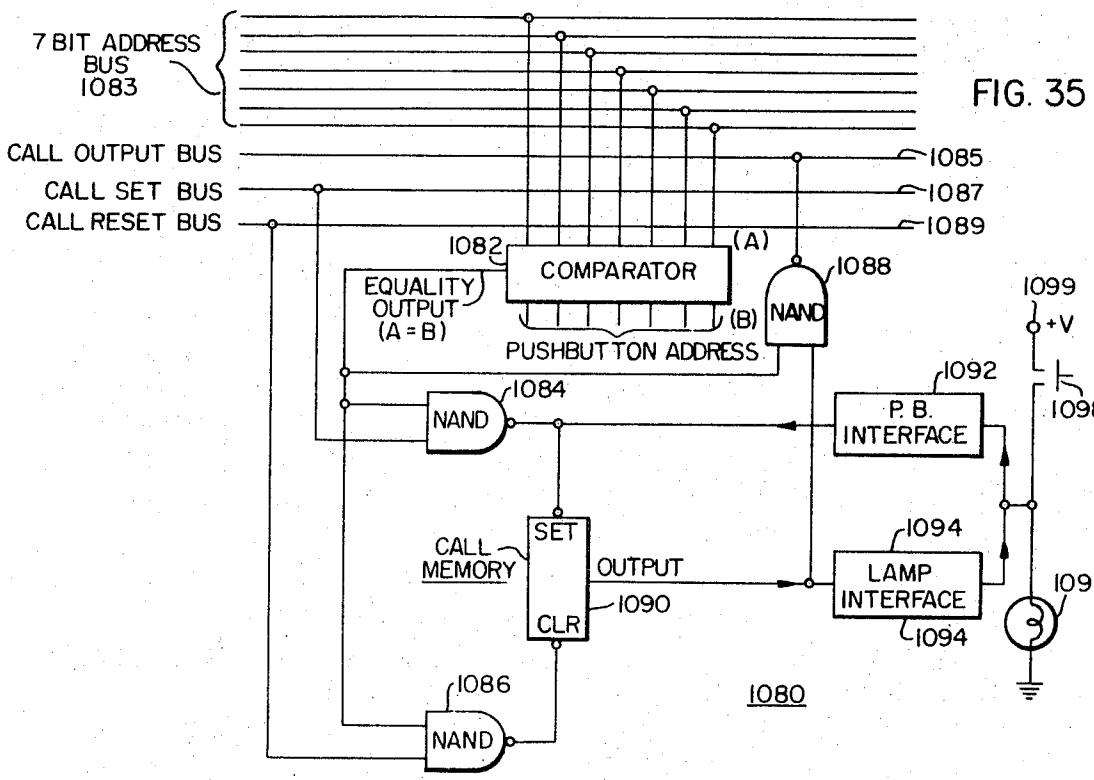


FIG. 35

FIG.33

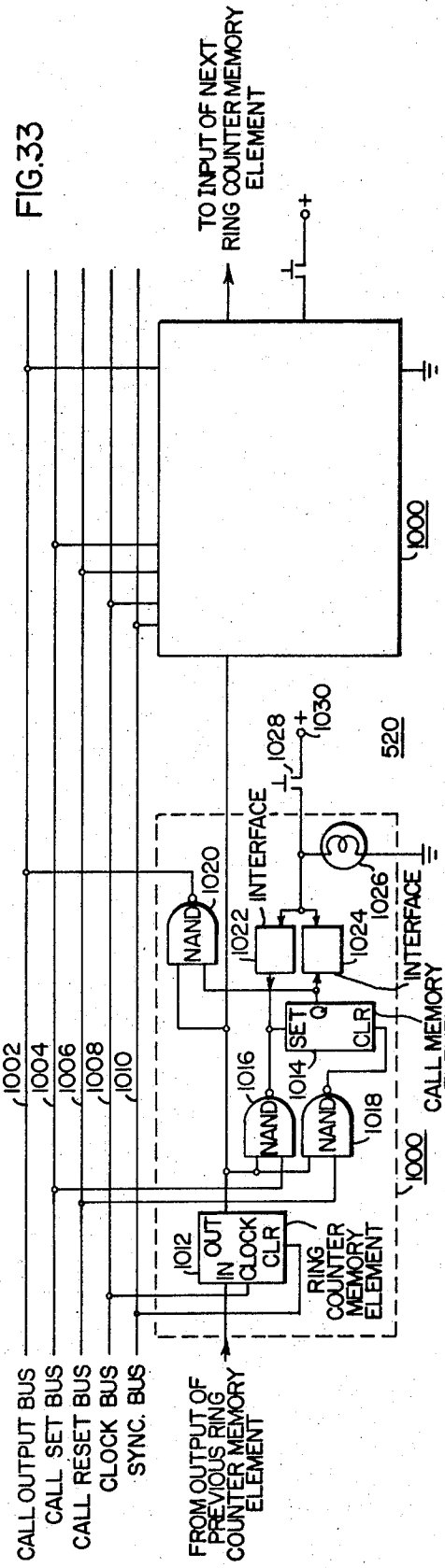
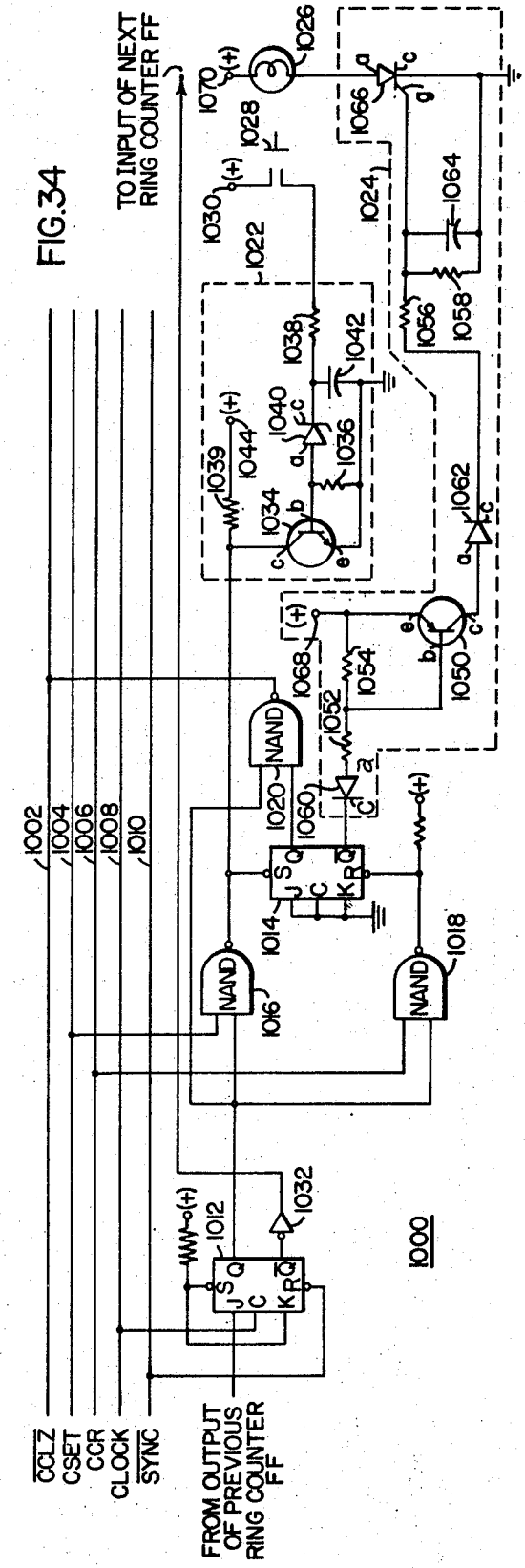

TO INPUT OF NEXT RING COUNTER MEMORY ELEMENT
1000
520
1028
1030
INTERFACE
1022
1024
1026
INTERFACE
CALL MEMORY ELEMENT
1020
NAND
SET
CLR
Q
1016
1014
1018
1012
OUT
IN
CLOCK
CLR
RING COUNTER MEMORY ELEMENT
1002
1004
1006
1008
1010
CALL OUTPUT BUS
CALL SET BUS
CALL RESET BUS
CLOCK BUS
SYNC. BUS
FROM OUTPUT OF PREVIOUS RING COUNTER MEMORY ELEMENT

FIG.34


TO INPUT OF NEXT RING COUNTER FF
1000
1070
1026
1028
1030
1022
1024
1064
1066
1056
1058
1038
1042
1039
1044
1040
1036
1034
1062
1068
1054
1050
1052
1020
1060
1014
1018
1016
1012
1032
1002
1004
1006
1008
1010
CCLZ
CSET
CCR
CLOCK
SYNC
FROM OUTPUT OF PREVIOUS RING COUNTER FF

ELEVATOR SYSTEM

CROSS-REFERENCES TO RELATED APPLICATIONS

Certain of the apparatus disclosed and described in this application, but not claimed, is claimed in the following concurrently filed applications:

Application Ser. No. 340,616, filed Mar. 12, 1973 which is assigned to the same assignee as the present application.

Application Ser. No. 340,620, filed Mar. 12, 1973 which is assigned to the same assignee as the present application.

Application Ser. No. 340,614, filed Mar. 12, 1973 which is assigned to the same assignee as the present application.

Application Ser. No. 340,618, filed Mar. 12, 1973 which is assigned to the same assignee as the present application.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates in general to elevator systems, and more specifically to an elevator system in which an elevator car has a main and an auxiliary car station therein for registering car calls.

2. Description of the Prior Art

Elevator systems having one or more cars for serving a plurality of landings in a building or structure often include two car stations in the elevator car to enable passengers to register their car calls at either side of the door. In the prior art, the pushbuttons of the two stations are connected in parallel, such that registering a call on either station will provide the same car call signal for use by the floor selector, and also will simultaneously actuate indicators on both car stations which indicate to the passenger that his call has been registered, and which notifies other passengers of the car calls which have been registered. The indicators are usually lamps, which are energized adjacent to or behind the pushbutton which is actuated, and also adjacent to or behind the corresponding pushbutton on the other car station.

When the elevator system is associated with a structure having relatively few landings, the parallel interconnection of the pushbuttons of the main and auxiliary car stations is not a great problem. However, when the associated structure has many landings, the parallel interconnection of pushbuttons requires a substantial amount of labor.

Briefly, the present invention is a new and improved elevator system which substantially reduces the labor required to electrically interconnect main and auxiliary car stations in an elevator car. Only a few wires are required between the stations, regardless of the number of landings in the associated structure.

More specifically, instead of interconnecting the pushbuttons of the main and auxiliary car stations in parallel, parallel interconnection is simulated by serializing the calls from each car station, and using the serialized cells from each station to set calls on the other station. The serialized car call signal from each station may be used to directly feed the car call set line of the other station, or since the two car call signals are combined to provide a single car call signal, the combined signal may be connected to the car call set line of each station. The car call set signal applied to the set line of each station will set a call on one station which was initiated on the other. The cycle time for serializing all of the calls, even for the tallest of structures, is such that the lamp on the car station the call is to be set on will be energized only a fraction of a second after the lamp is energized on the station used by the passenger to register his request. Thus, parallel interconnection of the pushbuttons of the two stations is simulated, eliminating the large number of wires usually required between the car stations, and the labor required to connect them.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention may be better understood, and further advantages and uses thereof more readily apparent, when considered in view of the following detailed description of exemplary embodiments, taken with the accompanying drawing in which:

FIG. 1 is a partially schematic and partially block diagram of an elevator system which may utilize the teachings of the invention;

FIG. 2 is a block diagram of an elevator system having a plurality of elevator cars, and a system processor which controls the cars by parallel inhibits;

FIG. 3 is a block diagram of an elevator system having a plurality of elevator cars, and a system processor which controls the cars by serial inhibits;

FIG. 4A is a block diagram of an elevator system having a plurality of elevator cars and a system processor which controls the cars by serial inhibits and "single address" interfacing between the system processor and floor selector;

FIG. 4B is a block diagram of an elevator system similar to the system of FIG. 4A, illustrating direct connection of corridor calls to the floor selector and a schematic diagram of inhibit logic which may be used with the single address interfacing concept;

FIG. 4C is a graph which illustrates the serial inhibits generated by the inhibit logic shown in FIG. 4B for all cars except a car assigned to answer a specific call;

FIG. 5 is a block diagram of an elevator system having a plurality of cars, and a system processor which controls the cars by serial inhibits and "family of the interfacing between the system processor and floor selector;

FIG. 5A is a block diagram of an elevator system which illustrates a modification of the system shown in FIG. 5;

FIG. 6 is a graphic representation of the family of call interfacing concept, using two floor address mode signals to interpret the commands from the system processor;

FIG. 7 is a block diagram of an elevator system wherein car calls are stored in each car, with the car calls and car signals being time multiplexed over the traveling cable to the car controller;

FIG. 8 is a block diagram of an elevator system in which an auxiliary car station is used in an elevator car, along with the main car station, with both the main and auxiliary car stations storing and serializing car calls registered thereon, and setting the registered calls on the other station with serial set lines which simulate parallel connections between the push buttons of the two stations;

FIG. 9 is a block diagram of an elevator system using dual rate time multiplexing, wherein a first data rate is

used between the car stations and their associated car controllers, and a second higher data rate is used between the car controllers and the system processor.

FIG. 10 is a graph which illustrates one transmission cycle of the low and high speed data links of the dual rate time multiplexed data transmission system shown in FIG. 9;

FIG. 11 is a block diagram of a new and improved elevator system which embodies the teachings of the invention;

FIGS. 12A and 12B are graphs which illustrate timing signals for generating complete scan slot cycles, and timing signals for addressing specific scan slots of the cycles, respectively, which signals are used in the elevator system shown in FIG. 11;

FIGS. 13A and 13B are graphs which illustrate timing signals developed in scan slot 000 of the scan cycles shown in FIG. 12A;

FIG. 14 is a block diagram of a floor selector which may be used in the elevator system shown in FIG. 11;

FIG. 15 is a schematic diagram of a call selector which may be used in the floor selector shown in FIG. 14;

FIGS. 16A and 16B are schematic diagrams of logic circuitry which may be used in the floor selector shown in FIG. 14;

FIGS. 17 and 18 are block diagrams of the input and output channels, respectively, of the system processor shown in the elevator system of FIG. 11;

FIGS. 19, 20, 21 and 22 illustrate the relationship between basic and high speed scan slots of various data links between the system processor and other functions of the elevator system shown in FIG. 11;

FIG. 23 illustrates an exemplary core map for the memory of the system processor of the elevator system shown in FIG. 11;

FIG. 24 is a block diagram of certain interface functions disposed between the system processor and each floor selector in the elevator system shown in FIG. 11;

FIG. 25 is a schematic diagram which may be used for the car call interface function shown in FIG. 24;

FIG. 26 is a schematic diagram of a corridor call register which may be used for this function shown in block form in FIG. 24;

FIG. 27 is a schematic diagram of call inhibit logic which may be used for this function shown in block form in FIG. 24;

FIG. 28 is a schematic diagram of circuitry which may be used for parking the elevator cars at specific landings in response to predetermined command signals from the system processor;

FIG. 29 is a schematic diagram of assignment logic which may be used for this function shown in block form in FIG. 24;

FIGS. 30A and 30B are block diagrams of certain multiplexing, demultiplexing and interface functions in the car station and car controller of the elevator system shown in FIG. 11;

FIG. 31 is a schematic diagram of a car call reset circuit which may be used in the elevator system shown in FIG. 11;

FIG. 32 is a schematic diagram of car call and reset circuitry which may be used in the elevator system shown in FIG. 11;

FIG. 33 is a schematic diagram of a car call memory and serializer which may be used in the elevator system of FIG. 11, illustrating the concept of storing and serializing car calls directly in the car call push button circuits;

FIG. 34 is a more detailed schematic diagram of the car call memory and serializer shown in FIG. 33, illustrating a specific implementation thereof which may be used; and

FIG. 35 is a schematic diagram of still another car call memory and serializer concept which may be used to store and serialize car calls from the car call push button circuits.

DESCRIPTION OF PREFERRED EMBODIMENTS

FIG. 1

Referring now to the drawings, and FIG. 1 in particular, there is shown an elevator system 10 which may utilize the teachings of the invention. Elevator system 10 includes a plurality of cars, such as car 12, the movement of which is controlled by a system processor 11. Since each of the cars of the bank of cars, and the controls therefor, are similar in construction and operation, only the controls for car 12 will be described in detail.

More specifically, car 12 is mounted in a hatchway 13 for movement relative to a structure 14 having a plurality of landings, such as 30, with only the first, second and thirtieth landings being shown in order to simplify the drawing. The car 12 is supported by a rope 16 which is reeved over a traction sheave 18 mounted on the shaft of a drive motor 20, such as a direct current motor as used in the Ward-Leonard drive system, or in a solid state drive system. A counterweight 22 is connected to the other end of the rope 16. A governor rope 24 which is connected to the top and bottom of the car is reeved over a governor sheave 26 located above the highest point of travel of the car in the hatchway 13, and over a pulley 28 located at the bottom of the hatchway. A pick-up 30 is disposed to detect movement of the car 12 through the effect of circumferentially spaced openings 26A in the governor sheave 26. The openings in the governor sheave are spaced to provide a pulse for each standard increment of travel of the car, such as a pulse for each .5 inch of car travel. Pick-up 30, which may be of any suitable type, such as optical or magnetic, provides pulses in response to the movement of the openings 26A in the governor sheave. Pick-up 30 is connected to a pulse detector 32 which provides distance pulses for a floor selector 34. Distance pulses may be developed in any other suitable manner, such as by a pick-up disposed on the car which cooperates with the regularly spaced indicia in the hatchway.

Car calls, as registered by push button array 36 mounted in the car 12, are recorded and serialized in car call control 38, and the resulting serialized car call information is directed to the floor selector 34.

Corridor calls, as registered by push buttons mounted in the corridors, such as the up push button 40 located at the first landing, the down push button 42 located at the thirtieth landing, and the up and down push buttons 44 located at the second and other intermediate landings, are recorded and serialized in corridor call control 46. The resulting serialized corridor call information is directed to the system processor 11. The system processor 11 directs the corridor calls to the cars through

an interface circuit, shown generally at **15**, to effect efficient service for the various floors of the building and effective use of the cars.

The floor selector **34** processes the distance pulses from pulse detector **32** to develop information concerning the position of the car **12** in the hatchway **13**, and also directs these processed distance pulses to a speed pattern generator **48** which generates a speed reference signal for a motor controller **50**, which in turn provides the drive voltage for motor **20**.

The floor selector **34** keeps track of the car **12** and the calls for service for the car, it provides the request to accelerate signal to the speed pattern generator **48**, and provides the deceleration signal for the speed pattern generator **48** at the precise time required for the car to decelerate according to a predetermined deceleration pattern and stop at a predetermined floor for which a call for service has been registered. The floor selector **34** also provides signals for controlling such auxiliary devices as the door operator **52**, the hall lanterns **54**, and it controls the resetting of the car call and corridor call controls when a car or corridor call has been serviced.

Landing, and leveling of the car at the landing, is accomplished by a hatch transducer system which utilizes inductor plates **56** disposed at each landing, and a transformer **58** disposed on the car **12**.

The motor controller **50** includes a speed regulator responsive to the reference pattern provided by the speed pattern generator **48**. The speed control may be derived from a comparison of the actual speed of the motor and that called for by the reference pattern by using a drag magnet regulator, such as disclosed in U.S. Pat. Nos. 2,874,806 and 3,207,265, which are assigned to the same assignee as the present application. The precision landing system using inductor plates and transformer **58** is described in detail in U.S. Pat. No. 3,207,265.

An overspeed condition near either the upper or lower terminal is detected by the combination of a pick-up **60** and slow-down blades, such as a slow-down blade **62**. The pick-up **60** is preferably mounted on the car **12**, and a slow-down blade is mounted near each terminal. The slow-down blade has spaced openings, such as a toothed edge, with the teeth being spaced to generate pulses in the pick-up **60** when there is relative motion between them. These pulses are processed in pulse detector **64** and directed to the speed pattern generator **48** where they are used to detect overspeeds.

A new and improved floor selector **32** for operating a single elevator car, without regard to operation of the car in a bank of cars, has been disclosed in co-pending application Ser. No. 254,007, filed May 17, 1972, now U.S. Pat. No. 3,750,850 which is assigned to the same assignee as the present application. In order to avoid duplication and to limit the complexity of the present application, application Ser. No. 254,007, filed May 17, 1972, is hereby incorporated by reference, and will hereinafter be referred to as the first incorporated application. Thus, only the changes in floor selector **32** which are necessary to adapt it to bank operation and control by a system processor will be described in detail.

FIG. 2

Dispatchers of the prior art conventionally direct the elevator cars of a bank to operate in one of a plurality of operating modes, with the specific operating mode depending upon traffic conditions. The cars are controlled in a predetermined pattern for each operating mode by signals generated by the dispatcher. When the dispatcher is inoperative for some reasons, an emergency dispatcher is used in order to send the cars away from the main terminal and thus continue to provide elevator service to the building.

FIG. **2** is a block diagram of a new and improved elevator system **100** which directs all the corridor calls to the floor selectors of all of the elevator cars of the bank. Control of the elevator cars in a predetermined pattern for a specific operating mode is accomplished by a system control means or system processor generating an inhibit signal for all of the cars which are not to answer a specific assigned corridor call. The car which is to answer the corridor call is not inhibited, and thus it responds in the manner in which it normally would respond in the absence of the system processor.

More specifically, FIG. **2** is a block diagram of a new and improved elevator system **100** which operates a bank of elevator cars by parallel inhibit signals. Elevator system **100** includes a plurality of cars, with two cars A and B, being illustrated. Car A includes a selector **104** and car call means **106**, and car B includes a selector **108** and car call means **110**. Corridor call means **112** provides an individual circuit for each corridor call, with four circuits **114, 116, 118**, and **120** being illustrated. For example, the lowest and uppermost floors would each have a single circuit for transmitting requests for up and down service, respectively, from these floors, and each intermediate floor would have two circuits, one for requests for down service and one for requests for up service, from that floor. These parallel corridor calls are directed to the system processor **102** via lines **114, 116, 118** and **120**, and also to each of the cars via conductors having like reference numerals except for the addition of a prime mark. The system processor **102** provides an output circuit for each corridor call circuit, for each elevator car of the bank. Thus, the processor **102** includes output circuits **122, 124, 126** and **128** for car A, output circuits **130, 132, 134** and **136** for car B, and corridor call circuits **114'**, **116'**, **118'** and **120'**. The floor selector of a specific elevator car can "see" a specific corridor call, unless the system processor has developed an inhibit signal for that car for that specific call. A corridor call on line **114** is directed to all of the cars, such as cars A and B, via circuit **114'**, and if the system processor directs car A to answer the call, inhibits are developed for the remaining cars. For example, an inhibit will be applied to the selector **108** of car B via circuit **130**. Car A, not receiving an inhibit signal over circuit **122** would process to answer the call.

An abnormal operating condition of the system processor **102**, as sensed by monitoring or emergency means shown generally at **138**, generates a signal $\overline{EMT}$ which removes all inhibits. Thus, in an emergency condition involving an abnormal operation of the system processor, each car will operate independently as described in the first incorporated application, whereby each car would answer all corridor calls requesting service in its travel direction, until there are no more calls for service in that direction, and then reverse its travel direction and answer all calls for service in the reverse direction until there are no further calls for service in that direction. All cars would respond to any specific

call for service in its travel direction, with the first car to arrive at the floor of the call serving the call.

The elevator system **100** of FIG. **2,** operating by inhibits, automatically places each of the cars on independent operation when the system processor fails to provide any inhibit signals. If the system processor functions abnormally while still providing inhibit signals, the detection of the abnormal condition results in the removal of all inhibits, and again the cars operate independently. The parallel inhibit system of FIG. **2** may be easily adapted to elevator systems of the prior art, requiring a system processor which generates inhibits signals for all cars except a car which is assigned to service a call, and a logic gate for each possible corridor call for each elevator car. Each corridor call is connected to an input of its own logic gate, along with an inhibit line from the system processor. The lack of an inhibit signal opens the gate, while an inhibit signal closes it.

The parallel inhibit arrangement of FIG. **2** necessarily requires a wire for each possible corridor call and an additional like number of wires for the inihibit signals, for each car of the elevator bank. This is not a problem when the structure served by the elevator system is a relatively small structure having a limited number of floors. When the structure has a large number of floors, the large number of wires required by the parallel inhibit system would add substantially to the installation cost of the apparatus.

FIG. **3**

FIG. **3** is a block diagram of a new and improved elevator system **140** which has the operating advantages of the elevator system **100** shown in FIG. **2,** without the disadvantage of requiring a corridor call line and an inhibit line per corridor call per car. More specifically, the elevator system **140** of FIG. **3** is a serial inhibit arrangement in which the up and down corridor calls provided by corridor call means **112** over lines **114, 116, 118,** and **120** are serialized by serializing or multiplexing means **142** providing serial up calls over line **144** and serial down calls over line **146,** with the calls for any specific floor of the associated building always appearing in the same time slot of a repetitive scan. The serial up and down corridor calls are distributed to the selectors of all of the cars of the bank, such as the selectors **104'** and **108'** of cars A and B, respectively, via an up call bus **148** and a down call bus **150.** The system processor **102'** generates inhibits for each call for all of the cars which are not assigned to answer the call, placing the inhibit signal in the proper scan slot. The serial inhibits tailored for each car are distributed to the selectors of all of the cars via inhibit buses, such as up and down call inhibit buses **152** and **154,** respectively, for the selector **104'** of car A, and up and down corridor call inhibit buses **156** and **158** for car B. The up corridor calls may be gated through a single gate for each car, and the down corridor calls may also be gated through a single gate for each car, with each of the two gates per car also having as an input an inhibit bus. In the absence of an inhibit signal from the system processor, the signal gate is open, allowing a serial corridor call to be processed by the selector of the associated elevator car. The remaining cars of the bank would have an inhibit signal present during the same time slot, closing the gates of these cars to this specific call. Failure of the system processor **102'** to produce inhibit signals automatically places the car on independent control. If the system processor is functioning, but abnormally, the detection of the abnormal condition by means **138** results in the generation of signal $\overline{EMT}$, which removes all inhibits and permits the selectors of the cars of the bank to "see" all of the calls.

While the corridor calls are illustrated in FIGS. **2** and **3** as proceeding to the floor selector of the elevator cars via the system processor, it is to be understood that the corridor calls may proceed directly to the floor selectors of the cars, as well as to the system processor.

The system processors **102** and **102'** shown in FIGS. **2** and **3,** respectively, may be a wired logic system, or a programmable system. A programmable elevator supervisory controller has the advantages of enabling different building sizes and configurations to be handled by the same controller hardware, as well as readily permitting reprogramming of the system processor with different strategy due to changed conditions in the building, or unsatisfactory performance of the originally provided strategy, or any condition which makes it desirable to change the strategy. This is due to the fact that the strategy of program is stored in the core memory of the processor, and is not in the hard wired logic. Programmable type systems permit spare part inventories to be reduced, since each programmable elevator system would use identical hardware. Expansion of a given elevator system would be possible on a modular basis, by reprogramming the system processor to take into account the additional car, or cars. Manufacturing lead time would be reduced because final details of options available to the purchaser would not have to be known before manufacture of the hardware. Malfunction diagnosis and repairs could be simplified by the use of a diagnostic program loaded either in the place of, or together with, the dispatching program. Training of field maintenance personnel would also be facilitated due to the fact that all of the hardware would be of like construction.

While the programmable system for elevator use has the advantages set forth above, the problem immediately arises of how to interface the programmable dispatcher or system processor, and the wired floor selectors of the various cars of the elevator bank. With wired logic dispatchers, this problem is not present.

FIG. **4**

FIG. **4A** is a block diagram of a new and improved elevator system **160** which solves the interfacing problem between a programmable system processor **162** and the selectors of a group of elevator cars, such as the selectors **104'** and **108'** of elevator cars A and B, respectively. Similar to the system shown in FIG. **3,** the elevator system **160** is of the serial type, placing the corridor call information into predetermined scan slots, with the information for a specific floor of the building to be served always occurring in the same scan slot. The inhibits are also of the serial type for each car.

More specifically, each floor of the building to be served is assigned a different binary number, and a scan counter **164** is provided which continuously scans or counts through these numbers. An assignment register is provided for each car, such as assignment registers **166** and **168** for cars A and B, respectively. When the programmable system processor **162** selects a specific car to answer a specific corridor call, it outputs to the selected car the floor address, i.e., the binary number assigned to the floor, of the call. This floor address, which will be called word A is applied to the assign-

ment register of the selected car and the outputs of the assignment registers **166** and **168** are applied to comparators **175** and **177**, respectively. The scan counter **164** is also connected to the comparators **175** and **177**. The output of the scan counter **164** will be referred to as word B. When word B is equal to word A in the comparator, the comparator outputs a signal which is applied to inhibit logic means, which then removes any inhibit signal it may have been generating for the time slot of the scan corresponding to the binary number of the selected corridor call. Thus, the corridor call for this time slot is gated into the floor selector of the car assigned to the call by the programmable system processor **162** to answer the corridor call. For example, cars A and B have inhibit logic means **170** and **172**, respectively, connected between their associated comparators and their floor selectors. If a down corridor call is registered from the sixth floor, and the system processor **162** assigns car A to answer this call, it will output the floor address of the sixth floor, and when the scan counter reaches this address, the comparator **175** will output an equality signal during this scan slot. This equality signal is applied to inhibit logic means **170** which removes any inhibit which it may have been generating for this scan slot, enabling car A to "see" this call. The assignment register for car B, on the other hand, will not be given the address of the sixth floor, and indeed, may be given the address of another floor to which the system processor has assigned this car. Further, the scan slot for the sixth floor for car B will be inhibited by an appropriate inhibit signal, preventing car B from seeing the sixth floor down call being answered by car A. In like manner, if car B has an assignment, car A will be inhibited from seeing the call being answered by assigned car B.

FIG. **4B** is a block diagram of an elevator system **160'** which is similar to the elevator system **160** shown in FIG. **4A**, except (a) modified to include a third car C, (b) modified to illustrate that the corridor calls may be directly sent to the floor selectors of the cars, and (c) including inhibit logic means which may be used to inhibit a call for all cars except an assigned car. Like reference numerals in FIGS. **4A** and **4B** refer to like functions.

The third car C includes an assignment register **169**, a comparator **179**, inhibit logic **174**, and a selector **198**, all interconnected with the system processor **162** and scan counter **164**, as hereinbefore described relative to cars A and B. The up and down corridor calls are applied directly to the floor selectors **104'**, **108'** and **198** via buses **148'** and **150'**, respectively.

The system processor **162** provides a signal $\overline{CCZ}$ which is low or true during each scan slot having a corridor call which has been assigned to a specific car. In other words, signal $\overline{CCZ}$ identifies all corridor calls which have been assigned. Signal $\overline{CCZ}$ is applied to each of the inhibit logic means **170**, **172** and **174**. Each comparator outputs a high or true A=B signal to its associated inhibit logic means when it has an assigned floor, and the scan counter **164** is presently at the associated scan slot of this floor. Thus, the output of a comparator identifies calls assigned to its associated car. Since each of the inhibit means is of like construction, only inhibit means **172** of car B is shown in detail.

Inhibit logic means **172** includes means for comparing signal $\overline{CCZ}$ with the output of comparator **177** and to provide inhibit signals for unassigned floors, More specifically, means **172** includes inverter or NOT gates **191** and **193**, and NAND gate **195**. The $\overline{CCZ}$ signal is applied to the input of inverter **191**, and the A=B signal is applied to the input of inverter **193**. The outputs of inverters **191** and **193** are connected to inputs of NAND gate **195**. The output of NAND gate **195** provides the inhibit signal $\overline{INHIBIT}$, which signal is applied to selector **108'**.

FIG. **4C** is a graph which aids in explaining the operation of the inhibit logic mean shown in FIG. **4B**. It will be assumed that down calls $\overline{2Z}$ are registered for the fourth, seventh, and 10 floors of a building served by elevator system **160'** and that these calls are assigned to cars A, B and C, respectively. The signal $\overline{CCZ}$ will thus be low or true during the scan slots of the fourth, seventh and 10 floors.

Signal $\overline{CCZ}$ will be high for floors not assigned to specific cars, forcing the output of NAND gate **195** high for these scan slots via inverter **191**. Calls located in these scan slot may thus be answered by any car which is in position to do so.

Signal $\overline{CCZ}$ will be low for floors assigned to specific cars. The NAND gate **195** of each inhibit logic means will thus have high inputs from inverter **191** for the fourth, seventh and 10 floors. The A=B signal from comparator **175** of car A will be high for the fourth floor, since car A has been assigned to this call, which signal is inverted by inverter **193** to force NAND gate **195** to provide a high output. The A=B signal from comparator **175** will be low for the seventh and 10 floors, which will be inverted to a "1" by inverter **193**. Thus, NAND gate **195** has two high inputs for floors seven and tens, providing a low or true inhibit signal $\overline{INHIBIT}$ for these floors.

In like manner, selector **108'** will receive true inhibit signals for the fourth and tenth floors, and selector **198** will receive true inhibit signals ro the fourth and seventh floors.

In summary, when the system processor **162** assigns a car to a specific corridor call, all cars except the assigned car will receive inhibit signals for the scan slot of the floor of the assigned call. This "specific car to a specific call" dispatching is especially useful for accommodating timed out calls, i.e., corridor calls which have been registered without being answered for longer than predetermined period of time.

FIG. **5**

FIG. **5** is a block diagram of a new and improved elevator system **180** which adds flexibility to the interfaceing concept shown in FIG. **4**. The "single word" communication between the system processor and the car controllers of the various elevator cars provide a simple interface but causes a duplication of effort and enlarges the program of the programmable system processor. For example, when an elevator car is not under control by the system processor it has the ability to answer a stack of corridor calls as it progresses through the building, as described in the first incorporated application. The contents of the stack of calls is contained in a series of memory elements which store the fact that a call is registered. When a simple "one word" interface is used, the stack of calls is duplicated within the system processor program, and a servicing program must be included which outputs the calls at the appropriate time and follows the progress of the various cars through the building to remove the calls from the stack as they are answered. This extra program takes core

space and increases the running time of the program.

Instead of assigning a car to answer a call on a one call at a time basis, the system of FIG. **5**, in addition to the options of FIG. **4**, enables the system processor to select an inhibit mode which enables a selected car to consider a selected family of calls with reference to a selected landing. The system of FIG. **5** includes a programmable system processor **182**, which in addition to the floor address word "A" outputs floor address mode signals to each car on buses shown generally at **184** and **186** for cars A and B, respectively, which signals determine the specific option selected by the system processor **182** for each car of the bank. The elevator system of FIG. **5** includes assignment registers for each car of the elevator bank, such as assignment registers **188** and **190** for cars A and B, respectively, which output the floor address word "A" to comparators **185** and **187**, respectively, which compare the output word "A" from the associated assignment register with the output word "B" from the scan counter **164**. The floor address word "A" is the reference landing. In addition to providing an equality signal when words "A" and "B" are equal, the comparator in the system of FIG. **5** also provides signals which indicate where the count of the scan counter **164** is at any instant, relative to the address of the reference floor (word "A"). For example, assume that word "A" provided by the programmable system processor **182** for car A is the binary address of the sixth floor. When the scan counter is counting through the first through the fifth floors, comparator **185** will provide a true A>B signal, indicating the scan counter **164** is scanning the floors below the assigned floor. When the counter **164** is outputting the binary number assigned to the sixth floor, the A = B signal will become true, and when the counter indexes to the seventh floor, the A<B signal will become true, and will remain true while the counter **164** counts through the remaining floors located above the sixth floor. The signals from the comparators of the cars are applied to inhibit logic means, such as inhibit logic means **192** and **194** for cars A and B, respectively, along with the floor address mode signals from the programmable system processor **182**. The inhibit logic means uses the floor address mode signals and signals from the comparator to interpret the command from the system processor, and provides serial inhibit signals in the proper time slots, i.e., for the proper floors, to carry out the command.

FIG. **6**

A functional implementation of the "family of call" concept set forth in the elevator system of FIG. **5** is shown in the graph of FIG. **6**. The functional implementation shown in FIG. **6** has the advantage of carrying out the commands of the programmable system processor **182** with only four basic signals. The first signal required is the floor address in binary. The number of bits in the floor address signal depends upon the number of floors in the building, and the number of scan slots used for purposes other than representing floors, such as to indicate zones. Since a seven-bit signal will provide 128 scan slots, it will be adequate for all known structures, and will be used for purposes of example. The floor address signal will be referred to as FADO–FAD6, to indicate the seven bit binary address. The next signals required are the floor address mode signals, which interpret the command from the system processor. Only two mode signals are required to interpret the commands, which signals will be referred to as MODO and MOD1. The final signal required is the service direction of the car, which may or may not be the same as the travel direction of the car. The service direction of the car may be obtained from a signal which will be referred to as UPSV. When UPSV is true, the car is set for up service, and when it is not true, or zero, the car is set for down service.

The chart shown in FIG. **6** indicates the calls a car is allowed to see or consider in the different floor address modes available to be selected by the system processor **182** of the elevator system **180** shown in FIG. **5**. The first mode is indicated by column **200**, which indicates a building having a plurality of floors, of which only 11 are shown, and also the values of th mode signals MOD0 and MOD1. In the mode illustrated by column **200**, both mode signals or bits MOD0 and MOD2 are zero or false. In this mode, it is immaterial what the floor address signal FAD0–FAD6 is, as all corridor calls are inhibited, indicated by the cross-hatching of all floors.

The second mode available to the system processor is indicated by column **202**. For this mode it will be assumed tha the floor address FAD0–FAD6 of the reference floor is that assigned to the fifth floor. This mode is selected by a true or high MOD0 signal and a false or zero MOD1 signal, and it results in all calls being inhibited except for the call at the addressed floor. Thus, all floors are cross-hatched except the fifth floor in column **202**. The service direction signal UPSV determines whether an up or down corridor call at the fifth floor is to be answered by the car, which car will be assumed to be at the first floor. If signal UPSV is true, (one) an up call at the fifth floor will be answered by the car, and if signal UPSV is false (zero), a down call at the fifth floor will be answered by the car.

The third mode available to the system processor **182** of the elevator system **180** shown in FIG. **5** is indicated by column **204**. This mode is selected by mode bit MOD0 being false, and mode bit MOD1 being true, and it will again be assuemd that the car is at the first floor and the floor address FAD0–FAD6 of the reference floor is that of the fifth floor. This mode enables the car to see or consider a predetermined family of calls, i.e., a call for service in a predetermined direction at the addressed floor and all calls for service in the same direction from all floors above the addressed or reference floor. Thus, only calls from floors one through four are inhibited, indicated by the cross-hatching of these floors in column **202**. Again, whether or not the car will see up or down corridor calls from the fifth floor and above depends upon the condition of the service direction signal UPSV. A true UPSV signal would allow the car to answer all UP calls from the floors, while a false UPSV signal would allow the car to answer down calls from the floors. In the latter instance, the car must travel up when set for down service, the car would go to the highest floor above the fifth floor which had a registered call for down service before reversing its travel direction. It would then service all down corridor calls down to nd including the fifth floor. If signal UPSV is true, the car would travel to the fifth floor before it would begin to service up corridor calls.

The fourth mode available to system processor **182** of FIG. **5** indicated by column **206**. The mode is selected by both mode bits MOD0 and MOD1 being true.

It will be assumed that the car is at the first floor, and that the floor address FAD0-FAD6 of the reference floor is that of the fifth floor. This mode also allows the car to "see" a predetermined family of calls, i.e., a call for service in a predetermined direction at the addressed floor, and all calls for service in the same direction from all floors below the addressed floor. A true UPSV signal would enable the car to recognize calls for up service from floors one through five, and being located at the first floor it would open its doors to accept a registered up call at the first floor. Even though the car may travel above the fifth floor to service registered car calls, it would cease to stop for up corridor calls once it had passed the fifth floor. A false UPSV signal would enable the car to recognize down corridor calls only in the zone which includes floors on through five. Since the car is located at the first floor, it would proceed to the highest call for down service registered at floors one through five, and then reverse its direction and service all down corridor calls as it proceeds back to floor number one.

In addition to modes which interpret a floor address, a mode may be selected which is not specifically tied to a reference floor. For example, a mode may be selected which enables the car to consider or see all calls in its assigned service direction within a predetermined zone. The zone may be predetermined by providing a read only memory programmed with the floors of the zone, which memory is enabled by the system processor. FIG. 5A is a schematic diagram of an elevator system **180'** which is similar to the elevator system **180** shown in FIG. **5**, except modified to enable the system processor **182** to select an inhibit mode not restricted to a reference floor. Like reference numerals in FIGS. **5** and 5A indicate like components.

More specifically, a programmable read only memory **173** is provided, and selector **104''** for car A is modified to include an additional gate for up corridor calls and an additional gate for down corridor calls. The selection for the remaining cars would also be so modified, and they would each include a read only memory, if they were also to have this zoning mode. For example, a NAND gate **181** would normally be provided in selector **104''** for screening up calls on call bus **148** with the output of inhibit logic **192.** Selector **104''** additionally includes an inverter gate **183,** connected to the output of NAND gate **181,** and a NAND gate **189,** which has one input connected to the output of inverter gate **183.** The output memory **173** is connected to the other input of NAND gate **189.** Memory **173** is programmed for the floors of the zone, and has inputs connected to the scan counter **164** and to the system processor **182.** The input to the read only memory **173** from the system processor is an "enable" input. Memory **173,** when enabled by the system processor **182,** outputs a serial signal which is low or logic zero for the scan slots representing floors not in the zone, and high or logic one for the scan slots representing floors in the zone. When memory **173** is not enabled, its output is high for all scan slots of the scan counter **164.** Thus, when the memory **173** is not enabled, NAND gate **189** has no screening effect, and the inhibit mode developed by inhibit logic **192** is effective.

When the system processor **182** enables the memory **173,** the system processor would also normally cause inhibit logic **192** to output a high signal to NAND gate **181,** to remove the screening effect of gate **181,** and all up calls would appear at the input of NAND gate **189** as a high signal. If the memory **173** is programmed for the floor of the call, NAND gate **189** will output a low or true signal $\overline{1Z}$. If the memory **173** is not programmed for the floor of the call, NAND gate **189** will have a low input from memory **173** for this scan slot, and the output of NAND gate **189** will go high.

FIG. 7

Car calls initiated from push buttons in the main car station must be stored, and indicators illuminated to indicate to the passengers the calls which have been registered. In elevator systems of the prior art, for each car call push button and lamp assembly in the car station, located in the car, there is a latching relay in the car controlled located in the penthouse. When a car call push button is depressed in the car station, its lamp is illuminated and a corresponding relay in the car controller is energized. This relay remains energized through its own latching contact, once it picks up, until the call is answered, which breaks the latching circuit of the relay. This prior art arrangement thus requires a wire from each push button and lamp assembly in the car to the relay in the car controller, via the traveling cable. The cost of wires in the traveling cable, including the factory and field labor to install then, is quite significant. Replacing the relays with solid state components does not solve this basic problem.

FIG. 7 is a block diagram of a new and improved elevator system **210** which substantially reduces the number of wires in the traveling cable, by an arrangement in which the car call memories are located in the car station. Car calls and car signals are time multiplexed from the car station to the car controller in the penthouse via the traveling cable, and resets for the car calls are time multiplexed from the car controller to the car station, via the traveling cable. Car call sets initiated by maintenance personnel in the penthouse are also multiplexed from the car controller to the car station. This arrangement works well with the serial inhibit concept illustrated in FIGS. **3, 4** and **5,** as the timing for controlling the multiplexing of the data between the penthouse and car station is already established.

More specifically, FIG. **7** illustrates a penthouse station for one or more elevator cars which includes a system processor **212,** and a floor selector, car controller and car call reset for each elevator car to be controlled by the system processor. In order to simplify the drawing, only a single car is shown in FIG. **7,** which car has a floor selector **214,** a car controller **216,** and car call reset means **218,** all located in the penthouse.

The single car also includes a car station **220** located in the elevator car itself, with the car station and car controller being interconnected via traveling cable **222.** The car calls are serialized in the car station **220,** an multiplexed, to the car controller, along with any car signals such as those generated by an attendant when a car is taken off of automatic control and operated by an attendant. The serial car call portion of the multiplexed signal is direction to the floor selector **214,** and when the car initiates deceleration to stop at a floor of a car call, this information is used to initiate a car call reset from car call reset means **218.** The car call resets are multiplexed from the car controller **216** to the car station **220.** This arrangement substantially reduces the number of wires required in the traveling cable, and

therefore reduces the installed cost of the elevator system.

The memory elements for car calls are disposed in the car station according to the concept of FIG. 7. Since the number of memory elements may vary between two, to over a hundred, depending upon the specific elevator system, the memory elements must be addable in a modulator arrangement to simplify the construction of the car station **220.** A number of small circuit cards could be mounted near the multiplexer, each of which include a number of memory elements and lamp driver interface elements. However, this approach would still be quite costly, due to the labor involved in connecting a wire from the correct push button to the correct module element, and then from the module element to the multiplexer. Thus, a building with a large number of floors, such as 100, would require 100 connection inputs to the multiplexer from the 100 memory elements. The memory elements and multiplexer, however, must be very compact, since they are installed in the car station in the concept shown in FIG. 7, and the space on the multiplexer required for these inputs alone would be prohibitively large.

FIG. 7 illustrates a new concept which solves the space and interconnection problems, by an arrangement which recognizes that the time multiplexing of data need not originate from the multiplexer which multiplexes the car calls and car signals to the penthouse, but the serialization of the car calls can originate in the push button circuits themselves. The memory element and associated time multiplexing and interface circuitry for each push button may be mounted directly behind, or even impregnated in each push button. The call output portions of the push button circuits are connected in parallel to a common data bus, and the push button circuits or modules would be accessed in a predetermined sequence to obtain an indication of which push buttons had been depressed. FIG. 7 illustrates this concept generally byillustrating a box **224** in the car station labeled push buttons, memories and serializer, along with a separate multiplexer **226** which combines or multiplexes car signals, indicated generally at **228,** with the serial car calls from push button circuits **224** for transmission to the car controller **216.** Two new and improved arrangements for accessing the push button circuits will be hereinafter explained in detail.

FIG. 8

FIG. 8 is a block diagram of a new and improved elevator system **230,** which is similar to that shown in FIG. 7, except it adds a new concept relative to auxiliary car call push buttons. Like reference numerals in FIGS. 7 and 8 refer to like components. When an auxiliary car station including an auxiliary push button station is added to an elevator car, enabling passengers on either side of the car to register their calls, the push buttons of the main and auxiliary car stations are connected in parallel. On tall structures, the labor required to connect the large number of wires between the auxiliary and main push button stations is substantial.

In the elevator system of FIG. 8, the push buttons of the main and auxiliary push button stations of the main and auxiliary car stations, respectively, are not paralleled. Elevator system **230** includes a car station **232** which includes a main car station having the push button circuits **224,** which include the call memory and call serializer functions, and an auxiliary car station which includes push button circuits **234,** which are similar to the main push button circuits **224.** The main and auxiliary push button circuits **224** and **234** each have their serial call lines connected to a multiplexer **236** which OR'S the main and auxiliary car calls to obtain total car calls for multiplexing with the car signals **228.** The multiplexer **236** also includes means for setting a car call registered on the auxiliary push button station in the main push button station, and vice versa, which set is transmitted to the push button circuits **224** and **234** via serial car call set lines. Setting a call in the main push button station which was registered on the auxiliary push button station, or setting a call on the auxiliary push button station which was registered on the main push button station, makes it appear that the two stations are connected in parallel, as associated call lamps are illuminated on both the main and auxiliary push buttons regardless of where the push button was actuated. The longest delay between the illumination of two associated call lamps would be about one-fourth of a second. Thus, the number of wires required between the main push button station, auxiliary push button station and multiplexer **236** are substantially reduced, compared with similar installations of the prior art, requiring only the reset, call and set lines illustrated between the push button stations and multiplexer, and clock and sync lines to properly operate the serial method of delivering calls, resets, and sets between the various functions of the car station.

The car call set signals on the serial set lines may simply be responsive to the car calls set on the other station, or they may be provided by combining the car calls from both the main and auxiliary car stations, and car call sets from the penthouse to provide a single serial car call set signal which is sent to both the main and auxiliary car stations. Car call resets may take precedence over sets by passing this single serial set signal through a gate which is blocked by a true reset signal, before sending the signal to the car stations.

FIG. 9

FIGS. 7 and 8 introduced the concept of locating the car call memories in each car station and multiplexing serial data between a car station located in the car and its associated car controller located in the penthouse. This arrangement substantially reduces the number of wires required in the traveling cable, reducing manufacturing and field labor accordingly

The interconnecting data link, i.e., traveling cable, between the car controller and the car station can be quite long, up to about 1,700 feet. The desired electrical isolation between wires of the traveling cable is difficult to achieve, since the electrical noise level over these cables can be very high due to the electrical power lines in the traveling cable which provide electrical power for car lighting, fans, and door operators.

FIG. 9 is a block diagram of a new and improved elevator system **450** which overcomes the electrical noise problem in the car to penthouse data link, without penalizing the supervisory functions of the system processor. In general, the elevator system of FIG. 9 includes first control means located in each elevator car, referred to as a car station, second control means located remotely from each car, such as in the penthouse, referred to as a car controller, and third control means, also located in the penthouse, referred to as a system processor. A first data link, referred to as a time multiplexed low speed link, interconnects each car station

and its associated car controller. A second data link, referred to as a time multiplexed high speed link, interconnects each car controller with the system processor. In general, first, second, third and fourth multiplexers are used to multiplex the data in the system, with the first and second multiplexers being located in the car stations and car controllers, respectively, and the third and fourth multiplexers located in the car controller and system processor, respectively. Specially, the elevator system **450** includes a system processor **452,** a plurality of elevator cars, such as cars A, B, C, and D, indicated by car stations **454, 456, 458** and **460,** respectively. Car controller **462, 464, 466** and **468** for cars A, B, C and D, respectively, are located in the penthouse. Corridor push buttons for registering up and down floor or landing calls are indicated generally at **470.** Circuitry for serializing the corridor calls from the corrdior push buttons **470** and transmitting the calls to the system processor **452,** is indicated generally at **472.** The car controllers are all linked with the system processor **452.** As indicated in FIG. **9,** the car controller **462** for car A may be combined with the corridor interface **472** in a package **474** which will be referred to as a master car controller.

The electrical noise problem in the traveling cable data link is solved, without hindering the system processor **452** in its decision making role, by a dual rate time multiplexing arrangement which time multiplexes data at two different rates. The data links between the car stations and their associated car controllers obtain the high noise immunity required by transmitting the intelligence signals at a relatively high voltage level e.g., 125 volts DC, and at a relatively slow data rate e.g., about 500 hertz. Using this relatively slow data rate and high voltage for the signals, conventional relay wires of the type used to interconnect electromechanical relays may be used in the longest of traveling cables. The 500 hz. data rate enables one car call and one car signal to be transmitted from a car station to its associated car controller every 2 milliseconds. In a 128 floor building, all calls would be transmitted every 256 milliseconds, which is adequate for car calls. It would serve no useful purpose to transmit car calls to the system processor faster than this rate.

This relatively slow data rate for car calls and car signals, however, does not hamper the monitoring and command functions of the system processor, as time multiplexed high speed links are used between the car controllers and the system processor, and between the corridor interface **472** and system processor. The interconnecting links between the car controllers and system processor is usually very short, since they are all located in the penthouse, enabling low voltage, high speed data links to be used. Only a small number of shielded pairs of wires are required in these high speed time multiplexed data links. Optical isolators or couplers may be used in these high speed data links to produce excellent noise immunity and electrical isolation.

FIG. **10**

FIG. **10** is a timing diagram illustrating a basic data transmission cycle, which, for purposes of example, will be assumed to be of 2 milliseconds duration. The basic cycle for the low speed links between the car stations and their associated car controllers is indicated by block **476,** illustrating a car call in one half of the cycle, and a car signal in the other half of the cycle. The basic cycle for the high speed links, between the car controllers and systems processor is indicated by block **478,** which, as illustrated, may operate at a rate of 8 khz. providing 16 bits of information during the basic cycle. The car call from the low speed link is included as one of the 16 bits of data in high speed link, with monitoring and command signals occupying the other 15 locations of the basic transmission cycle.

Table I

The inhibit concept of dispatching, hereinbefore described, which has the advantage of automatically placing the elevator cars on independent operation, as described in the incorporated application, should the system processor fail to provide inhibit signals, may be advantageously applied to other functions of an elevator system. Table I illustrates how first and second mode bits, which are referred to as signals HLM1 and HLM2, respectively, may be used to override normal hall lantern operation at the option of the system processor.

TABLE I

TRUTH TABLE FOR HALL LANTERN MODE BITS

HLMI	HLM0	Definition
1	1	Normal Operation
0	0	Inhibit illumination of both lanterns
1	0	Turn on down hall lantern
0	1	Turn on up hall lantern

The logic levels of signals HLM0 and HLML for normal operation of the hall lanterns will depend upon the logic levels of the signals when the system processor is inoperative. If, for example, signals HLMO and HLML are at the logic **1** level when the system processor is inoperative, this combination would be used to permit normal operation. Then, as illustrated in Table I when both signals HLMO and HLM1 are at the logic zero level, the system processor would override normal hall lantern operation by inhibiting the illumination of both the up and down hall lanterns. When signal HLMO is a logic zero, and HLM1 is a logic one, the system processor would override normal operation and direct that the down hall lantern for a selected floor be illuminated. When signal HLMO is a logic **1** and signal HLM1 is a logic zero, the system processor would override normal operation of the hall lantern and direct that the up hall lantern for a selected floor be illuminated.

The system processor also overrides normal door operation in manner which automatically permits normal operation in the event the system processor is inoperative or otherwise out of service. For example, the system processor may provide a signal $\overline{\text{DCLO}}$ which overrides normal door opening when true. When the system processor is inoperative, signal $\overline{\text{DCLO}}$ is false or at the logical one level permitting normal door opening. Further, the system processor may provide a signal $\overline{\text{DOPN}}$ which overrides normal door closing when true. When the system processor is inoperative, signal $\overline{\text{DOPN}}$ is false or at the logical one level, permitting normal door closing.

FIGS. **11, 12** and **13**

FIG. **11** is a block diagram of an elevator system **500** which embodies the concepts hereinbefore set forth, except for the parallel inhibit concept of FIG. **2.** The elevator system of FIG. **11** uses the serial inhibit concept with dual rate time multiplexing of data, as this ar-

rangement substantially reduces the number of wires required in the system, and thus reduces the manufacturing and field labor required to manufacture, install, and maintain the elevtator system.

More specifically, elevator shown **500** includes a plurality of elevator cars, each of which include a car station mounted in the elevator car itself, and a car controller mounted in the penthouse. Since the car stations are all similar, only a single car station **502** is shown in the Figure, which is for a car referred to as car A. Since the car controller for car A may be combined with corridor call control, while the remaining car controllers are independent of the corridor call control, two car controllers are illustrated, a master car controller **504** for car A, and a car controller **506,** which is for a car referred to as elevator car B.

The master car controller **504** is connected to the up and down corridor call push buttons, which are indicated generally by reference numeral **507.** Although the up and down corridor calls are indicated as being collected on common buses **509** and **511,** respectively, it is not necessary to store and serialize the corridor calls in the push button circuits, such as hereinbefore described relative to the car calls. The corridor calls do not go to the penthouse in a traveling cable, nor is the space required to connect a large plurality of wires to a multiplexer a problem in the penthous. Thus, the system **500** will be described as one in which the corridor calls are brought to the master car controller in parallel. However, the call memory and serialization concept described relative to car calls may be applied to corridor calls, if desired.

Elevator cars A and B, as well as any remaining cars of the bank, may operate independently as described in the first incorporated application, with the floor selectors **508** and **510** shown in FIG. **11** corresponding to the floor selector **34** of the incorporated application.

Elevator system **500** includes a programmable system processor **512** for controlling the movement of the bank of elevator cars according to traffic conditions and the programmed strategy in response thereto. The elevator system **500** may be programmed, for example, to follow the strategy outlined in U.S. Pat. Nos. 3,256,958 and 3,292,736, which are assigned to the same assignee as the present application, or any other suitable strategy may be desired. Concurrently filed application Ser. No. 340,619, filed Mar. 12, 1973 in the name of James Vine, entitled "Elevator System" which is assigned to the same assignee as the present application, discloses a programmable system processor which may be used for system processor **512.** This application is hereby incorporated by reference and will be hereinafter referred to as the second incorporated application.

The system processor **512** controls the movements of the various elevator cars in the bank by serial inhibits, with the portion of the floor selector **34** of the incorporated application which are modified to accept serial inhibit control being hereinafter described.

In describing the elevator system **500** shown in FIG. **11,** it will be helpful to set forth the various signals and their definitions which will be referred to relative to FIG. **11,** and also when describing figures which will be referred to when various functions illustrated in block form in FIG. **11** are described in greater detail.

SIGNAL	FUNCTION
ACALLS	Serial car calls from Auxiliary to Main car station
ACC	Car to accelerate or travel at full speed
$\overline{\text{ACCY}}$	Acceleration request — remains true until DEC goes true
$\overline{\text{ACCX}}$	Acceleration request — similar to $\overline{\text{ACCY}}$
$\overline{\text{ASET}}$	Car call set for auxiliary car station
$\overline{\text{ATDN}}$	Attendant down signal
$\overline{\text{ATSV}}$	Car on attendant service
$\overline{\text{ATUP}}$	Attendant up signal
AVAS	Car is available at selector level
AVP0–AVP6	Advanced car position in binary
BOTTOM (BTTM)	Car is within 18 inches of bottom terminal
$\overline{\text{BSMT}}$	Basement Assignment
$\overline{\text{BYPS}}$	Car bypassing corridor calls
CA	Call above floor of advanced car position
CALL	Car call registered
CALLS	Combined car calls from main and auxiliary car stations
CB	Call below floor of advanced car position
CCAB	Car call above advanced car position
$\overline{\text{CCAI}}$	Inhibits actual car calls from being answered
CCBL	Car call below advanced car position
$\overline{\text{CCLZ}}$	Serial car calls in main car station
$\overline{\text{CCR}}$	Car call reset
$\overline{\text{CCS}}$	Car call set
$\overline{\text{CCST}}$	Car call strobe
$\overline{\text{CCY}}$	Serial car calls
$\overline{\text{CCZ}}$	Serial floor assignments
$\overline{\text{CEN}}$	Call enable
$\overline{\text{CFLY}}$	Call at floor during first scan
$\overline{\text{CIN}}$	Corridor call answer inhibit
$\overline{\text{CLPB}}$	Car signal indicating a car call pushbutton has been pushed
$\overline{\text{CR}}$	Car call reset
CREG	Car call registered
CREN	Corridor call enable
$\overline{\text{CRSM}}$	Car call reset
$\overline{\text{CSET}}$	Car call set for main car station
$\overline{\text{CSST}}$	Car signal strobe
$\overline{\text{CSZ}}$	Serialized car signals
DCE	Down call enable
$\overline{\text{DCL}}$	Doors closed
$\overline{\text{DCL}\phi}$	Close car doors — processor command
$\overline{\text{DCY}}$	Serial down calls
$\overline{\text{DEC}}$	Deceleration request
$\overline{\text{DECS}}$	Synchronized deceleration request
$\overline{\text{DNC}}$	Serialized down corridor calls
$\overline{\text{D}\phi}$	Door open request enable
$\overline{\text{D}\phi\text{PN}}$	Open car doors — processor command
$\overline{\text{DOR}}$	Door open request
$\overline{\text{DORR}}$	Reset-door open request
DNRZ	Serial down corridor call reset
DNSV	Down service direction
DORS	Door open request suppress
DRCL	True when all doors closed
DNSV	Down service direction
D45	Requests door closure when a logic one
E1	Call at advanced car position
$\overline{\text{E1X}}$	Request for stop
$\overline{\text{EMT}}$	True when system processor is out of service — places cars on through trip operation
EQIR	Serial car position — (unstrobed)
$\overline{\text{EQ1Z}}$	Serial car position (Strobed)
$\overline{\text{EQ2}}$	Advanced car position equal to floor level
FAD0–FAD6	Floor address
$\overline{\text{FCC}}$	Forced car call for parking purposes
$\overline{\text{FEN}}$	Floor enable
$\overline{\text{FSC}}$	First Scan
$\overline{\text{HLD}}$	Enable for down hall lanterns
$\overline{\text{HLL}}$	Hall lantern enable
HLM0	Hall lantern mode bit 0
HLM1	Hall lantern mode bit 1
HLU	Enable for up hall lanterns
HLX	Hall lantern enable
IDLE	Car idle, ready to make run
INSC	Car is in service with system processor
IWO–IWZ	Input words to system processor

LCC	Clock signal
LSC	Sync signal
LC1, LC3, LC5, LC8	Data signals
LOAD N	Load car position counters
LW0–LW–LW2	Load words 0, 1 and 2
MAIN	Car within 18 inches of main floor
MCDA	Serial car data
MCR	Master car call reset
MNFL	Main floor start signal from processor
MOD0	Floor address mode bit 0
MOD1	Floor address mode bit 1
MT00	Memory track
MT01	Memory track
MXVM	Maximum velocity
NC	Logic one when no calls
NCL	Logic one when no calls
NCS	Logic zero when no calls
NEXT	Car next
NLC	Input pulses to counter 544
OW0–OW2	Output words from system processor
PARK	Park command from system processor
PCR	Indicates advanced car floor position has changed
PD	Pulse advanced car position counter down
PFL	Parking floor assignment
PCCR	Higher voltage CCR signal
PCCS	Higher voltage CCS signal
PREAD	Higher voltage READ signal
PWRIT	Higher voltage WRIT signal
PSZ	Serialized penthouse signals
PU	Pulse advanced car position up
READ	Car station data (car calls and car signals)
REN	Ring counter enable — (from memory track)
RUN	Run signal — selector can see a call
SAC	Scanning above advanced car position
SASS	Service assignment
SBC	Scanning below advanced car position
SC	Clock signal
SDS	Set selector for down service
SDT	Set selector for down travel
SLDN	Car slowing down
SS	Sync signal
STT	Special through trip
SUS	Set selector for up service
SUT	Set selector for up travel
S0S–S6S	Scanning signals
TASS	Travel Assignment
TDS	Terminal stop signal
TOP	Car within 18 inches of top floor
TRI	Serial data from system processor
TSD	Car on terminal slowdown when logic 0
UCE	Up call enable
UCY	Serial up corridor calls
UPC	Serial up corridor calls
UPRZ	Serial up corridor call reset
UPTR	Up travel
UPSV	Up service direction
WRIT	Multiplexed EQIR and Penthouse signals
WT50	Car load greater than 50 percent of full capacity
WT75	Car load greater than 75 percent of full capacity
1Z	Serial up corridor calls
2Z	Serial down corridor calls
3Z	Serial car calls
32L	Signal from relay which is true when car is moving

Also, before describing the elevator system **500,** it will be helpful to illustrate some of the serial timing waveforms used to properly sequence the master car controller **504,** car controller **506,** and car stations **502,** as well as the flow of data to and from the system processor **512.**

FIGS. **12A** and **12B** are timing diagrams which illustrate how call information pertaining to different floors of the building to be served by the elevator system is monitored sequentially in time. Each floor of the building is assigned a specific count of a scan counter, such as a binary counter located in each car controller, referred to as slave timing **514** in FIG. **11.** The outputs of this scan counter are referred to as S0S through S6S in FIG. **12A.** This seven-bit counter repetitively divides successive predetermined time intervals into a plurality of scan or time slots. For example, the counter may be pulsed once every 2 milliseconds, thus making a complete scan of its 128 possible states in 256 milliseconds. Each floor of the building is assigned to a particular count of this counter, with the counts being referred to as scan or time slots. Each floor thus has a specific scan slot, but each scan slot does not necessarily designate a floor level. For example, some scan slots may be used to indicate when a car is in an express zone. Also, some scan slots may be unused.

Depending upon the number of floors and the number of scan slots required indicating car position in an express zone, the scan counter will be programmed to provide 16, 32, 64 or 128 scan slots, as indicated in FIG. **12A,** before returning the count to all zeros.

The seven-bit scan counters in the car controllers are maintained in synchronizm by a master scan counter, indicated by the master timing block **516** in the system processor **512,** which provides master clock and sync signals LCC and LCS, respectively, to each car controller.

The slave timing **514** also generates waveforms used for decoding a particular scan slot. Sector signals SEC0 through SEC7 shown in FIG. **12B** are generated from scan counter bits S4S, S5S and S6S by one of eight decoder. Each sector signal is true for 16 scan slots, and they are thus used to select a specific group of 16 scan slots. Eight scanning address signals **SA01** through **SA15,** using the odd numbers from 1 through 15 to indicate the eight signals are generated from scan counter bits S0S through S3S by another one of eight decoder. Each of these scanning address signals repeats during each of the sector signals. Thus, any odd numbered scan slot of a sector may be selected by AND-ing the desired sector signal with the desired scanning address signal. For example, AND-ing SEC0 with **SA09,** provides scan slot **9** of the first sector. As illustrated in FIG. **11,** the master car controller **504** contains a corridor call memory **518,** with individual memory elements being addressed by the sector and scanning address signals. It is not necessary to decode the even numbered scan slots, as the serial data appearing in the even numbered scan slots may be stored an addressed during the next odd numbered scan slot. However, the even numbered scan slots may be decoded and addressed during their time slots, if desired.

FIGS. **13A** and **13B** illustrate the signals which occur in one time slot, with the first time slot of a scan, referred to as time slot **000,** being illustrated as certain synchronization signals indicated by an asterisk in FIGS. **13A** and **13B** only occur during the first time slot of each scan cycle. The highest frequency timing signal provided by slave timing **514** is a 32 khz signal referred to as **K32.** Signal K32 is generated in response to an external clock signal SC, derived from signal LCC generated by the master timer **516.** Should the slave timer fail to receive the external timing signal SC, an internal multivibrator will provide the **K32** signal. However, in this event these 32 Khz signals in the various car controllers would not be synchronized with the rest of the

system, and detecting this failure of the master timing signal each car will operate independently on what may be called block operation, handling real car calls and dummy car calls which direct each car to stop at a different block of floors, in order for all of the floors of the building to be equitably served during this malfunction of master timing.

The K32 clock signal is subdivided by binary counters to produce signals K16, K08, KO4, KO2, K01 and KP5. The KP5 signal, in another divider chain, creates the scan counter bits S0S through S6S, hereinbefore described. Signals K08S, K02S, K01S and KP5S are developed by shifting the signals K08, K02, K01 and KP5, respectively, forward by 90°, which shifted signals are used by the floor and the car stations. Signal $\overline{SS}$ occurs only during scan slot 000, and is the synchronization input to the slave timer 514. $\overline{SYNCS}$ is used to maintain synchronism between the scan counter in the slave timer 514 and a scan counter located in the car station 502.

Signals STA, STAS, STB, STBS, $\overline{STC}$, S100, S200, S300 and S4 are all timing strobes utilized by the floor selector and call processing circuitry. As indicated by the asterisks, signals S100, S200 and S300 only occur during scan slot 000.

Each scan slot is subdivided into 16 high speed scan slots, giving the addresses HA00 through HA15, as indicated at the bottom of FIG. 13B. Signals FSTA and FSTB are data strobes occuring during each of the high speed scan slots, and are used in the time multiplexed dual data rate transmission concept shown in FIGS. 9 and 10, to transfer data over the high speed data links. Signal FST0 is used for resetting the high speed data transfer circuits.

Signals CSST and CCST are car signal and car call stobes for strobing car signals and car calls from the multiplexed signal READ sent up from the car station to its car controller. The signal READ is referred to as PREAD when it is at the higher voltage level for transmission over the traveling cable. As illustrated in FIG. 13B, car calls occupy the first half of signal READ and car signals the last half of the slot. Signal 3Z is the serial car call signal, which is strobed from signal READ by signal CCST.

Signal WRIT, which is called signal PWRIT at the high voltage level, is sent from the car controller to the car station, and it includes the advanced car position signal EQIR in the first half of the time slot, and signals PSZ which are generated in the penthouse.

Signal CCR, which is PCCR at the high voltage level, contains a clock enable signal REN during the first half of the time slot, which signal will be hereinafter explained, and car call reset signal CRSM during the last half of the scan slot. Signal PCCR is transmitted from the car controller to the car station.

Signal CCS, referred to as PCCS when it is at the high voltage level, is the car car set signal, which is transmitted from the car controller to the car station when maintenance personnel wish to set a car call in the car station.

Signals RSYNCO, RSYNCOA, and ASYNC are synchronization signals for synchronizing the serialization of car calls from the main and auxiliary push button stations in the car station 502.

Signals RCLOCKO, RCLOCKOA and ACLK are clock signals for clocking data from the main and auxiliary push button circuits.

Signals CCLZ and CCLZA are serial car calls from the main and auxiliary push button circuits, respectively.

Signal ACALLS are serial car calls from the auxiliary car station to the multiplexer, which multiplexes car calls and car signals to the car controller.

Signals CRESO, CRESOA, and ARES are car call reset signals responsive to signal $\overline{PCCR}$.

Signals CSET CSETA and ASET are car call set signals, which may originate from signal PCCS, or between the auxiliary and main push button stations.

Referring again to the elevator system 500 shown in FIG. 11, the car station 502 includes a main push button station 520 which performs the functions of storing and serializing the car calls registered by passengers in the car. These calls are synchronized by signal RSYNCO and clocked out in serial form by signal RCLOCK, appearing as signal $\overline{CCLZ}$ at the output of the main push button station 520. The synchronization and clock signals RSYNCO and RCLOCKO are responsive to sync and clock signals $\overline{PSYNCS}$ and PKO1S provided by the car controller 504.

An auxiliary push button station 522, when used, performs the same functions as the main push button station 520. Auxiliary push button station 522 stores car calls registered on the auxiliary push buttons, and provides serial car calls signals $\overline{CCLZA}$ at its output in response to synchronization and clock signals RSYNCOA and RCLOCKOA, respectively, provided by auxiliary car station control 524. The auxiliary car station control 524 is responsive to synchronization and clock signals $\overline{PSYNCS}$ and $\overline{PKO1S}$ provided by the master car controller 504. The auxiliary car station control 524 converts the serial auxiliary car calls $\overline{CCLZA}$ to serial auxiliary car calls $\overline{ACALLS}$, as shown in FIG. 13. The serial car calls $\overline{CCLZ}$ and $\overline{ACALLS}$ from the main and auxiliary push button circuits are applied to car call and car signal control 526. Control 526 notes car calls from the main and auxiliary push button stations and provides the car call set signals ASET and CSET, in response thereto, respectively, to illuminate the registered car calls on both the main and auxiliary push button stations, regardless of their origin. Control 526 receives car call reset signals $\overline{PCCR}$ from the master car controller 504 and sends reset signals CRESO and ARES to the main and auxiliary push button circuits, respectively. Control 526 also includes a multiplexer which provides signals $\overline{PREAD}$ by multiplexing the serial car calls with the car call signals originating in the car station. The car signals, for example, include signal $\overline{CLPB}$ which is generated when any car call push button is depressed, and signals which are generated by an attendant, such as an attendant up, an attendant down, and attendant by-passing signals, when the car is under attendant control.

The car call and car signal control 526 also includes a demultiplexer and display for receiving signal $\overline{PWRIT}$ from the car controller and demultiplexing the signal to obtain signal EQIR, tha advanced car position signal, and penthouse signals, such as car direction. These signals are displayed in the car station.

The car call and car signal control 526 is connected to car and penthouse signal control 530 in the master car controller 504 via a traveling cable 532, with the signal sent over the traveling cable being at the relatively slow transmission rate of about 500 hz, and at a relatively high voltage, about 125 volts DC, in order to

obtain the necessary immunity from electrical noise. The car and penthouse signal control **530** serializes the penthouse signals providing a signal $\overline{PSZ}$ which is applied to slave timing **514.** Slave timing **514,** in addition to providing the timing signals hereinbefore described, includes a multiplexer for combining the advanced car position signal EQIR with the penthouse signals $\overline{PSZ}$ to provide a multiplexed signal referred to as $\overline{WRIT}$. Signal $\overline{WRIT}$ is sent back to the car and penthouse signal control **530** where its voltage level is increased and sent to the car station **502** as signal $\overline{PWRIT}$.

Control **530** reduces the voltage level of signal $\overline{PREAD}$ and demultiplexes the signal to the extent of removing the car signals from the car calls. The car calls, still in serial form are sent to the car call reset control **534** via a signal $\overline{READ}$ (1), denoting the first half of the signal $\overline{READ}$, and the car signal portion, which will be referred to as signal $\overline{READ}$ (2) is sent to slave timing **514,** which provides serial car signals $\overline{CSZ}$. The serial car signal $\overline{CSZ}$ are sent back to control **530,** which demultiplexes them for use in the penthouse. For example, the car signal $\overline{CLPB}$ is sent to a multiplexer **536,** which will be hereinafter explained.

Control **530** receives car call reset signal $\overline{CCR}$ from car call and reset control **534,** it increases the voltage level of the signal, and sends the signal to the car station **502** as signal $\overline{PCCR}$.

Car call and reset control **534** receives the car call signal $\overline{READ}$(1) from control **530,** and sends serial car call signals 3Z to the floor selector **508.** Control **534** receives reset signal $\overline{CR}$ from car call reset **538,** and sends the car call reset to control **530** as signal $\overline{CCR}$.

Car call reset **538** receives the advanced car position signal EQIR, and the acceleration request signal $\overline{ACCX}$ from floor selector **508.** When signal $\overline{ACCX}$ goes false or high, indicating a deceleration request has been made for the floor of the advanced car position signal EQIR, a car call reset signal $\overline{CR}$ is generated.

Up and down corridor calls are generated at the various floors via push buttons **507,** and these signals are collected by the corridor call memory **518** via the up and down buses shown generally at **509** and **511,** respectively. Corridor call memory **518** stores the up and down corridor calls, and serializes them, placing them in the proper scan slots, using the timing signals SA0-1–SA15 and SEC0–SEC7, generated by slave timing **514.** The serial up and down calls $\overline{UPC}$ and $\overline{DNC}$, respectively, are applied to a corridor call multiplexer **540,** which multiplexes the up and down calls along with the floor enable signal FEN from a suitable memory track. The multiplexed up and down corridor calls, and floor enable signal FEN, are sent to the system processor **512** via link LC1 at the high transmission rate, hereinbefore described relative to the dual rate time multiplexing concept of FIG. **9.** Corridor call multiplexer **540** receives its timing from master timer **516** via clock and sync signals LCC and LCS. These clock and sync signals are then sent to multiplexer **536.** In the remaining car stations, the timing for multiplexer **536** comes direct from the master timer **516.**

The system processor **512** includes interface circuitry **542,** which is connected to core memory and processing functions **544** and **546,** respectively. The processing function **546** is connected to an input/output interface **548,** which in turn may be connected to a tape reader **550.** The specific strategy to be used by elevator system **500,** as well as the specifics of the associated building, are transferred from paper tape via the tape reader **550,** input/output interface **548,** and processor **546,** to the core memory **544.**

Interface **542** of the system processor **512** receives data via a high speed link LC1 from the master car controller **504,** which link includes up and down corridor calls, and a high speed link LC5 from the master car controller, as well as a high speed link LC5 from each of the remaining car controllers. High speed link LC5 includes several 16-bit input words, which include such signals as car calls $\overline{CCY}$, car call above CCAB, car call below CCBL, up corridor call reset $\overline{UPRZ}$, down corridor call reset $\overline{DNRZ}$, up service UPSV, door closed DRCL, car available AVAS, advanced car position AVP0–AVP6, slow down SLDN, in service $\overline{INSC}$, and up travel UPTR. The data in these high speed time multiplexed signals is received from the floor selector of the associated car, such as floor selector **508** of car A.

The system processor **542** assembles the data received via the high speed links LC1 and LC5 and makes decisions in accordance with its programmed strategy. Up and down corridor call reset signals $\overline{UPRZ}$ and $\overline{DNRZ}$ are sent to the corridor call reset demultiplexer **552** via the high speed link LC3, and serial up and down car call reset signals $\overline{UPRZ}$ and $\overline{DNRZ}$ are transmitted to corridor call memory **518,** which reset the call memories when the call is being answered.

Commands from the system processor **512** are sent to multiplexer **536** of each car controller via high speed links LC8. The high speed links LC8 contain the necessary signals for creating the serial inhibits required to operate the cars of the elevator system according to the programmed strategy. Multiplexer **536** provides a serial data signal TRI which is derived from link LC8, which signal is applied to a demultiplexer and assignment logic function **554.** The serial data TRI from function **536** is loaded into the multiplexer in function **554,** in response to load signals $\overline{LW0}$, $\overline{LW1}$ and $\overline{LW2}$ provided by function **536.** The assignment logic of function **554** is responsive to call above and below signals CA and CB, respectively, car available signal AVAS, car call above and below signals CCAB and CCBL, respectively, and up travel signal UPTR, all provided by the associated floor selector, such as floor selector **508.**

The multiplexer **554** sends the serial up and down corridor calls 1Z and 2Z, respectively, to the associated floor selector, along with command and inhibit signals, such as corridor call inhibit $\overline{CIN}$, set selector for up travel $\overline{SUT}$, set selector for down travel $\overline{SDT}$, emergency through trip $\overline{EMT}$, special through trip $\overline{STT}$, car call inhibit $\overline{CCAI}$, open door $\overline{DOPN}$, close door $\overline{DCLO}$, the car is next signal $\overline{NEXT}$, and hall lantern mode signals HLM0 and HLM1.

FIG. **14**

The floor selector described in the first incorporated application will control a car when it is operating independently. On independent control, the car will answer all calls ahead of its travel direction, and when there are no further calls for service in its travel direction, it will consider calls for service in the opposite direction. For example, when traveling downwardly, the car will answer all down corridor calls ahead of it, starting at the floor of its advanced car position, and when there are no down calls it will travel to the lowest registered up call. When it reaches this call, it will reverse its

travel direction and travel upwardly, answering all up calls until there are no further up corridor calls to service. The car will then travel to the highest registered down call, reverse and handle down calls. While the car will not necessarily travel from terminal to terminal in this independent mode, this independent mode will be referred to in this application as "through trip."

FIG. 14 is a block diagram of floor selector 508 shown in FIG. 11, which is similar to the floor selector 34 shown in FIG. 3 of the first incorporated application, except modified where necessary to accept serial inhibit control from a system processor, such as programmable system processor 512. The discreet advanced car position in binary signal AVP0–AVP6 required by multiplexer 536 is available at the output of reversible counter 72, and the serial advanced car position signal EQIR is available from comparator 82, which is essentially the same signal as EQIZ, except EQIZ is a strobed signal, and is of slightly shorter duration than the unstrobed signal EQIR. The only significant changes required in the circuitry of the floor selector of the incorporated application are required in the call selector 92 and logic 96, and thus these functions are given the same reference numerals, except for the prime marks in FIG. 14, to indicate that they have been modified.

FIG. 15

FIG. 15 is a schematic diagram of a call selector 92' which will perform the functions required of block 92' in the block diagram of the floor selector 508 shown in FIG. 14 call selector 92' is similar to call selector 92 of the first incorporated application, except modified for operation with a programmable system processor. More specifically, call selector 92' includes additional input terminals for receiving signals $\overline{\text{DECS}}$, D45, $\overline{\text{DOPN}}$, $\overline{\text{AVAS}}$, $\overline{\text{PCR}}$, $\overline{\text{HLX}}$, CREN, UPSZ, $\overline{\text{CIN}}$, DNSV, INSC, HLM0, HLM1, $\overline{\text{FCC}}$, $\overline{\text{CCAI}}$, STA and $\overline{\text{DCLO}}$. The terminals are identified by the same letters as the signals applied thereto.

Signal $\overline{\text{DECS}}$ is from synchronizer 94 of the floor selector 509, and it is logic zero or true when a request is made to decelerate the elevator car to stop at a landing. Signal D45 is from the door circuits, which request the doors to close when it is a logic one. Signal $\overline{\text{DOPN}}$ is from the system processor 512, and it is a logic zero when the system processor requests that the doors be opened. Signal $\overline{\text{AVAS}}$ is from logic 96' of floor selector 508, and it is a logic zero when the car has answered all its calls and is available for assignment. Signal $\overline{\text{PCR}}$ is from synchronizer 94, and it is a logic zero when the serial advanced car position changes. Signal $\overline{\text{HLX}}$ is a hall lantern enable signal from logic 96'. Signal CREN, which may be from a memory track, is high or true when the floor of the scan slot is enabled for corridor calls. Signals UPSV and DNSV are up service and down service signals, respectively, from logic 96', defining the service assignment which is set by the system processor 512. Signal $\overline{\text{CIN}}$ is a serial corridor call inhibit signal from the assignment logic function 554. Signal INSC is true or logic one when the car is in service, under control of the system processor 512. Signals HLM0 and HLM1 are hall lantern mode signals generated by the system processor 512, which signals override normal hall lantern operation when so directed by the system processor, and as hereinbefore explained with reference to table I. Signal $\overline{\text{FCC}}$ is true when the system processor 514 issues a command to park. Signal $\overline{\text{CCAI}}$ is generated by the system processor 512, when car calls are to be inhibited. Signal STA is a strobe signal, as illustrated in FIG. 13. Signal $\overline{\text{DCLO}}$ is a "close door" command from the system processor 512.

Call selector 92' requires additional NAND gates 560, 562, 564, 566, 568, 570, 571 and 573, additional inverter or NOT gates 572, 574, 576, 578, 580 and 582, and flip-flops 586 and 592. Flip-flops 586 and 592 may be of the cross coupled NAND gate type, with flip-flop 586 having cross coupled NAND gates 588 and 590, and flip-flop 592 having cross coupled NAND gate 594 and 596.

Signals $\overline{\text{DECS}}$ and D45 are connected to inputs of NAND gate 594 of flip-flop 592, signal $\overline{\text{DOPN}}$ is connected to an input of NAND gate 596 of flip-flop 592, signals $\overline{\text{AVAS}}$ and $\overline{\text{PCR}}$ are connected to inputs of NAND gate 588 of flip-flop 586, signal $\overline{\text{HLX}}$ is connected to an input of NAND gate 590, signal CREN is connected to inputs of NAND gates 252 and 262, signals UPSV and DNSV are connected to inputs of NAND gates 252 and 262, respectively, signal $\overline{\text{CIN}}$ is connected to inputs of both NAND gates 252 and 262, signal INSC is connected to inputs of NAND gates 560, 562 and 564, signal HLM0 is connected to inputs of NAND gate 566, 570 and 276, signal HLM1 is connected to inputs of NAND gates 566, 568 and 278, signal $\overline{\text{FCC}}$ is connected to an input of NAND gate 571, signal $\overline{\text{CCAI}}$ is connected to input of NAND gates 248 and 272, signal STA is connected to an input of NAND gate 573, and signal $\overline{\text{DCLO}}$ is connected to inputs of NAND gates 296, 297 and 301.

The output of NAND gate 252, instead of being directly connected to inverter 254, is connected to inverter 254 via inverter 572 and NAND gate 560. The output of NAND gate 262, instead of being directly connected to inverter 264, is connected thereto via inverter 574 and NAND gate 562. Signal UCE, instead of being directly connected to NAND gate 276, is connected thereto via inverter 580 and NAND gate 568. Signal DCE, instead of being directly connected to NAND gate 278, is connected thereto, via inverter 582 and NAND gate 570. Instead of connecting output signal $\overline{\text{DOR}}$ directly to inverter 302, it is now connected to an input of NAND gate 596 of flip-flop 592. Signal DCL is additionally connected to an input of NAND gate 596 of flip-flop 592, and the output of NAND gate 295 of flip-flop 294 is additionally connected to an input of NAND gate 590 of flip-flop 586. The output of NAND gate 588 of flip-flop 586 is connected to an input of NAND gate 566, and the output of NAND gate 594 of flip-flop 592 is connected to an input of NAND gate 564. The outputs of NAND gates 568, 570, 564 and 566 are respectively connected to an input of NAND gate 276, an input of NAND gate 278, to inverter 302, and to an input of NAND gate 564. The outputs of inverters 304 and 308 are additionally connected to output terminals $\overline{\text{HLU}}$ and $\overline{\text{HLZ}}$ via inverters 576 and 578, respectively, and the direct connections of output terminals $\overline{\text{HLU}}$ and $\overline{\text{HLD}}$ to the outputs of NAND gates 306 and 310, respectively, are removed. NAND gate 571 replaces inverter 270, with car call signals $\overline{3Z}$ being connected to an input thereof, and the output of NAND gate 571 is connected to inputs of NAND gates 248 and 272. NAND gate 573 has its remaining inputs connected to receive signals CEN, EQIZ, and the output of NAND gate 571. The output

of NAND gate 573 is connected to an input of NAND gate 280.

In the operation of call selector 92', the up and down corridor calls $\overline{1Z}$ and $\overline{2Z}$, respectively, in addition to being screened by the memory track signals MT00 and MT01, are also affected by signals UPSV, DNSV, CREN, $\overline{CIN}$ and INSC.

Signal CREN is an additional corridor call enable signal, which must be at the logic 1 level for each scan slot corresponding to a floor in order for the corridor calls to pass the gates 252 and 262. The corridor call enable signal CREN is the wired OR of the MTOO and MTO1 memory tracks, and is necessary for the condition when the car is at a terminal floor. Signals UPSV and DNSV introduce the assigned service direction of an elevator car, which is not necessarily the same as travel direction. Thus, before a car is able to consider up or down corridor calls, it must be on up service (UPSV = 1) or down service (DNSV = 1), respectively. Further, if the corridor call inhibit signal $\overline{CIN}$ is low or true for a specific scan slot, NAND gates 252 and 262 will block any corridor call for that scan slot. The system processor 512 controls the movement of a bank of elevator cars by providing true inhibit signals for elevator cars which are not to respond to specific corridor call locations. A car which may answer a specific corridor call or a car which is assigned to a specific corridor call receives a high or false $\overline{CIN}$ signal for the time slot of the call, enabling the corridor call to proceed through NAND gate 252, or NAND gate 262. The service direction assignment signals UPSV and DNSV determine whether the car will answer an up call or a down call at a specific floor for which it is not inhibited. If the system processor 512 becomes inoperative, or is otherwise taken out of service, the inhibit signal $\overline{CIN}$ automatically will be high, enabling the corridor calls to be considered by all cars without interference by the system processor.

If the car is deliberately taken out of service such that it is not under control of the system processor 512, signal INSC goes to the logic O level, blocking NAND gates 560 and 562, and preventing the car from "seeing" corridor calls. However, the car will still respond to car calls $\overline{3Z}$.

Car calls $\overline{3Z}$ are actual calls for service registered from the car call push buttons in the car. Signal $\overline{FCC}$ is a "dummy" car call, and is located in a different portion of the scan slot than a real car call $\overline{3Z}$. A dummy car call $\overline{FCC}$ is strobed by strobe signal STA, and thus appears in the second quarter of a scan slot, while actual car calls exist during the later S4 portion of the scan slot. Dummy car calls are used by the system processor 512 to park the cars at predetermined locations, with the parking command signal $\overline{PARK}$ and the parking floor location signal PFL both being generated in response to the system processor 512, which signals in turn generate the forced or dummy car call signal $\overline{FCC}$.

Signal $\overline{CCAI}$ is also generated by the system processor 512. When signal $\overline{CCAI}$ is true, NAND gates 248 and 272 are both inhibited or blocked, preventing the car from answering its car calls.

The signals which control operation of the hall lanterns are UCE and DCE, the up and down corridor call enables, respectively, the hall lantern mode signals HLMO and HLM1 from the system processor 514, which may override normal operation of the hall lanterns, and the output signals of the newly added flip-flops 586 and 592. Flip-flop 586 may be called the hall lantern enable flip-flop, and flip-flop may be called the hall lantern timing flip-flop, because of their respective functions.

First assume that signals HLM0 and HLM1 are both at the logic one level, which permits normal operation of the hall lanterns. If the output of NAND gate 295 of flip-flop 294 goes to zero, indicating a stop is being made, output $\overline{HLL}$ of flip-flop 586 goes to logic zero, and the output of NAND gate 566 goes to logic one, enabling NAND gate 564 and the hall lanterns. The same sequence is followed if signal $\overline{HLX}$ from logic 96' becomes true, indicating that there are calls ahead. Flip-flop 586 is reset by signal $\overline{PCR}$ when the car moves from the floor. If the car does not move from the floor, flip-flop 586 is reset by signal AVAS when the car becomes available.

The hall lantern timing flip-flop 592 is set by signal $\overline{DECS}$, which goes to logic zero when the elevator car starts to decelerate to stop at a landing. If the car is standing at the floor when the call comes in, signal D45 will go to logic zero and this sets flip-flop 592. When the doors start to close at the end of the non-interference time, signal $\overline{DECS}$ goes to logic one, signal D45 goes to logic one, and if there are no requests for the doors to reopen, flip-flop 592 is reset. When there are no requests for the door to open, signal $\overline{DOR}$ from the call selector 92' and signal $\overline{DOPN}$ from the system processor 512 are both logic one's and signal DCL is a logic one, indicating the doors may be considered closed.

When the hall lantern mode signals HLM0 and HLM1 are both "one's" and the car is enabled for up corridor calls, signal UCE will be a "one" and signal DCE will be a "zero." IF conditions are as just described with the hall lantern flip-flop 586 set, enabling the hall lanterns, and flip-flop 592 is set, controlling the timing of the hall lanterns, the output of NAND gate 564 goes to zero, which applies logic one's to the inputs of NAND gates 276 and 278 via inverter 302. NAND gate 276 is enabled by the true UCE signal while NAND gate 278 is blocked by the false DCE signal. The output of NAND gate 276 thus goes low, enabling the up hall lantern via a low signal $\overline{HLU}$. Signal $\overline{HLD}$ on the other hand will be high or false. If the call selector is enabled for down corridor calls instead of up corridor calls, signal $\overline{HLD}$ will become true via NAND gate 278, and NAND gate 276 would be blocked.

If the system processor 512 wishes to inhibit the operation of both the up and down hall lanterns, mode signals HLM0 and HLM1 will both go to logic zero, and both NAND gates 276 and 278 will be blocked. Thus, the hall lantern enable signals $\overline{HLU}$ and $\overline{HLZ}$ are both on the logic one level.

If the system processor 512 wishes to turn on the up hall lantern, hall lantern mode signals HLM0 and HLM1 will be a logic one and a logic zero, respectively. NAND gate 568 will output a logic one, regardless of the logic level of the up call enable signal UCE. NAND gate 566 will output a logic one regardless of the condition of the hall lantern enable flip-flop 586, and true signal $\overline{DOPN}$, also provided by the system processor 512, will force flip-flop 592 to output a logic one to NAND gate 564, regardless of whether or not the call selector 92' is requesting the door to open. Thus, NAND gate 376 has three high inputs, providing a true

$\overline{HLU}$ signal. NAND gate **278** is blocked by the low signal HLM1, and signal $\overline{HLD}$ is thus high.

If the system processor **512** wishes to turn on the down hall lantern, the hall lantern mode signals HLM0 and HLM1 will be a logic zero and a logic one, respectively. In this situation, NAND gate **276** is blocked, while NAND gate **278** receives all high inputs in the manner explained when describing the forced lighting of the up hall lantern, forcing signal $\overline{HLD}$ low.

The basic control of the up and down corridor call resets is derived from the hall lantern control signals $\overline{HLU}$ and $\overline{HLD}$. The advanced serial car position EQIZ gates a true $\overline{HLU}$ or a true $\overline{HLD}$ signal through NAND gates **306** and **310,** respectively, generating up and down corridor call reset signals $\overline{UPRZ}$ and $\overline{DNRZ}$, respectively, in the proper time slots.

Since the system processor **512** operates the bank of elevator cars by serial inhibits, allowing a car to move by not inhibiting a certain corridor call, some means must be provided for enabling the system processor **512** to move the cars around when necessary to place them at predetermined floor locations as directed by the programmed strategy. As hereinbefore explained, this is accomplished by generating dummy car call signal $\overline{FCC}$. However, when the car is being moved to a location in response to a dummy call, it will not be desirable to turn on a hall lantern, or to open the car door. It will be noted that NAND gate **248** only picks out "real" car calls by virtue of the S4 strobe of NAND gate **240,** and thus only real car calls set flip-flop **294,** the output of which controls the hall lantern enable flip-flop **586** and the door open flip-flop **298.** NAND gate **573** picks out dummy car calls due to strobe STA, and its output is connected to NAND gate **280** which will issue the request to stop signal $\overline{EIX}$ for dummy car calls as well as actual car calls and corridor calls. Thus, the car will move to the proper floor location and stop when under the influence of a dummy car call, but the hall lanterns will not be illuminated and the doors will not open.

FIGS. **16A** and **16B**

FIGS. **16A** and **16B** are logic diagrams which may be used for the logic function **96'** shown in FIG. **14.** FIGS. **16A** and **16B** include the modifications and additions which are required to adapt logic **96** of the first incorporated application to control by the system processor **512.** Like reference numerals in FIGS. **16A, 16B** and FIG. **9** of the incorporated application, refer to like components.

FIG. **16B** adds the logic necessary to develop up and down service direction signals UPSV and DNSV, respectively, when the elevator car is under control of the system processor, and the logic necessary to develop car call above and car call below signals CCAD and CCBL, respectively, for use by the system processor. These signals when true, indicate a car call above, or below, respectively, the serial advanced car position.

FIG. **16A** illustrates the modifications which must be made to logic **96** of the first incorporated application to introduce the changes brought about by the introduction of the assigned service direction, as well as to provide signals SLDN, $\overline{AVAS}$, and NCL for use by the system processor. Signal SLDN is true from the time the car starts to decelerate until the doors start to close. Signal $\overline{AVAS}$ is true when the car is available to accept an assignment from the dispatcher. Signal NCL is true when the selector can see no calls.

Referring now to FIG. **16B,** the system processor **512** provides service direction signals $\overline{SUS}$ and $\overline{SDS}$ which, when true, request that the floor selector be set for up service and down service, respectively. The service direction logic for processing these requests, includes a service direction flip-flop **600** having cross coupled NAND gates **602** and **604,** car call above flip-flop **606** having cross-coupled NAND gates **608** and **610,** car call below flip-flop **612** having cross-coupled NAND gates **614** and **616,** NAND gates **618, 620, 622, 624, 626, 628, 630, 632,** and **634,** inverter or NOT gates **636, 638, 640, 642, 644** and D-type flip-flops **646** and **648.**

The up and down service request signals $\overline{SUS}$ and $\overline{SDS}$ from the system processor **512** are connected to inputs of NAND gates **602** and **604,** respectively, of the service direction flip-flop **600.** The output of NAND gate **602** of flip-flop **600** is connected to an input of NAND gate **624.** The output of NAND gate **624** is connected to a terminal which provides the signal DNSV, and to an input of NAND gate **622.** The output of NAND gate **622** is connected to a terminal which provides the signal UPSV. Output signals UPSV and DNSV are the up and down service signals, respectively, which are connected to the portion of logic **96'** shown in FIG. **16A,** and to the call selector **92'** shown in FIG. **15.** When signal $\overline{SUS}$ is true, requesting the floor selector to be set for up service, NAND gate **602** outputs a logic one, NAND gate **624** outputs a logic zero, assuming its other inputs are also at the logic one level, which is a false DNSV signal, and NAND gate **622** outputs a true up service signal UPSV. The service direction flip-flop **600** may also be set to provide a true up service signal UPSV by input signal $\overline{ATUP}$, which is set by an attendant when the car is under control of an attendant. The service direction flip-flop **600** may also be set by a car call ahead when the car is traveling upwardly, with this function being provided by the output of NAND gate **618,** which has an input connected to receive UPTR, which is true when the car travel direction is up, an input connected to the output of NAND gate **608** of the car call above flip-flop **606,** and an input connected to receive timing signal **S200** which occurs on scan slot **000** at the start of each scan cycle.

The service direction flip-flop **600** may be set to provide a true down service signal DNSV by a true $\overline{SDS}$ signal from the system processor, by a true signal $\overline{ATDN}$ provided by an attendant, and by a car call ahead when the car is traveling downwardly. The latter condition is established by NAND gate **620,** which has inputs connected to receive signal UPTR via inverter **636,** an input connected to the output of NAND gate **614** of the call above flip-flop **612,** and an input connected to receive timing signal **S200.**

The call above flip-flop **606** has an input of NAND gate **608** connected to the output of NAND gate **626.** NAND gate **626** provides a low output during timing signal S4 when the scan counter is scanning above the position of the serial advanced car position (SAC = 1) and encounters a car call ($\overline{CCY} = 0$). The call above flip-flop **606** is reset by timing signal **S300,** but a car call above signal is stored for the complete scan cycle by D-type flip-flop **646,** providing a stored car call above signal CCAB for use by the system processor **512.** The $\overline{Q}$ output of flip-flop **646** is connected to output terminal CCAB via inverter **642.** When the output

of NAND gate **608** goes high, signifying a car call above, flip-flop **646** is triggered by the positive edge of signal **S200**, providing a low $\overline{Q}$ output and thus a high CCAB signal. The high output of NAND gate **608** switches the output of NAND gate **618** low at signal **S200** if the car is set for up travel (UPTR = 1), triggering flip-flop **600** to provide a low down service signal DNSV and a high up service signal UPSV from NAND gates **624** and **622**, respectively.

The call below flip-flop **612** has an input of NAND gate **614** connected to the output of NAND gate **628**. NAND gate **628** provides a low output during timing signal **S4** when the scan counter is scanning below the position of the serial advanced car position (SBC = 1) and encounters a car call ($\overline{CCY}$ = 0). The call below flip-flop **612** is reset by timing signal **S300**, but a car call below is stored for the complete scan cycle by D-type flip-flop **648**, providing a stored car call below signal CCBL for use by the system processor **512**. The $\overline{Q}$ output of flip-flop **648** is connected to output terminal CCBL via inverter **644**. When the output of NAND gate **614** goes high, signifying a car call below, flip-flop **648** is triggered by the positive edge of signal **S200**, providing a low $\overline{Q}$ output and thus a high car call below signal CCBL via inverter **644**. The high output of NAND gate **614** switches the output of NAND gate **620** low at signal **S200** if the car is set for down travel (UPTR = 0), triggering the flip-flop **600** to provide a high down service signal DNSV and a low up service signal UPSV from NAND gates **624** and **622**, respectively.

As will be hereinafter explained, a car call ahead of the direction of car travel will automatically override a request by the system processor to set the service direction in the opposite direction, automatically providing signals $\overline{SUS}$ and $\overline{SDS}$ which are consistent with the travel direction.

NAND gates **624** and **622** each have additional inputs connected to receive signals $\overline{STT}$, $\overline{EMT}$ and INSC. Signal $\overline{STT}$ is true when the car is on special service, such as on basement and top extension service, signal $\overline{EMT}$ is true when the system processor **512** is out of service, and signal INSC is true when the car is in service with the system processor. Thus, when the car is on special service ($\overline{STT}$ = 0), the system processor is not operational ($\overline{EMT}$ = 0), or the car is not in service with the system processor (INSC = 0), both NAND gates **622** and **624** are forced to output true up service and down service signals UPSV and DNSV simultaneously, and the service direction is thus not a consideration. In other words, the car operates as described in the incorporated application, answering all calls in its travel direction and then reversing and answering calls for service in the opposite direction.

In addition to the service direction signal UPSV and DNSV, FIG. **16B** develops signals for the portion of logic **96'** shown in FIG. **16A**, in response to the car service direction and car travel direction. NAND gate **630** has inputs connected to receive the service direction signals UPSV and DNSV, and as long as one of these signals is at the logic 0 level, NAND gate **630** outputs a logic 1, enabling NAND gates **632** and **634**. NAND gate **632** has an input connected to receive the up travel direction signal UPTR via inverter **640**, and NAND gate **634** has an input connected to receive the up travel direction signal UPTR. Thus, when the service direction is being used, NAND gate **632** outputs a logic one on up travel, and a logic zero, on down travel, and NAND gate **634** outputs a logic zero on up travel and a logic one on down travel. The outputs of NAND gates **632** and **634** are connected to inputs of NAND gates **326** and **332**, respectively, of the portion of logic **96'** shown in FIG. **16A**.

Additional changes to FIGS. **16A** include connecting signals UPSV and DNSV to inputs of NAND gates **364** and **366**, respectively, connecting the attendant up signal $\overline{ATUP}$ and the attendant down signal $\overline{ATDN}$ to inputs of NAND gates **352** and **354**, respectively, of the travel direction flip-flop **350**, providing a slowdown signal SLDN for the system processor which is true from the time the car starts to decelerate to stop at a floor, until the door starts to close, and providing a signal $\overline{AVAS}$ which is true when the doors are closed (DCL = 1), the car is not in the process of decelerating to stop at a floor (DEC = 0), and the car has answered all its calls ($\overline{RUN}$ = 1).

The up call enable and down call enable signals UCE and DCE, respectively, are thus responsive to the service direction, when the service direction function is operational. In other words, when the service direction function is operational, both the travel and service direction must be proper before the car will stop to answer a corridor call. For example, when the car is traveling up, the call enable signal UCE will only be true if the car is on up service. If the car is on down service and traveling up, the signals UCE and DCE will both be false until the car reverses its travel direction. Then, the down call enable signal DCE will become true. When the service direction function is not operational, both of the service direction signals UPSV and DNSV will be logic one's and the up and down call enable signals will be responsive to travel direction.

Signals $\overline{SUT}$ and $\overline{SDT}$ are signals provided by the system processor, which initially sets the car travel direction. Signal $\overline{SUT}$ requests that the selector be set for up travel, and signal $\overline{SDT}$ requests that the selector be set for down travel. Signals $\overline{ATUP}$ and $\overline{ATDN}$ are added to the input of travel direction flip-flop **C50**, to enable an attendant to set the travel direction.

Signals $\overline{SUT}$ and $\overline{SDT}$ initially set the travel direction, and the travel direction is then controlled by signals CA and CB from the call above and call below flip-flops **330** and **332**. When the system processor is not operational, or the car is not under control of the system processor, travel direction is entirely under control of signals CA and CB, or through attendant control.

The additional input to NAND gate **326** from FIG. **16B** blocks a call above from triggering the call above flip-flop **330** when the car is traveling downwardly, and the service direction function is operative.

The additional input to NAND gate **332** from FIG. **16B** blocks a call below from triggering the call below flip-flop **332** when the car is traveling upwardly.

If the service direction function is not operative, both inputs to NAND gates **326** and **332** from FIG. **16B** will be logical one's and the call above and call below flip-flops **330** and **332** will operate as described in the first incorporated application.

A signal $\overline{AVAS}$, which indicates the car is available to accept an assignment, is developed by a NAND gate **650**, which has inputs connected to the output of inverter **396** ($\overline{DEC}$), to the output of NAND gate **410** ($\overline{RUN}$), and to a newly added input terminal DCL, which provides a true signal when the doors are closed.

Thus, the car is available when the doors are closed (DCL = 1), it is not stopping at a floor, ($\overline{DEC}$ = 1), and it has answered all its calls ($\overline{RUN}$ = 1).

The inverter **323** of the incorporated application which has its input connected to the output of NAND gate **374** ($\overline{ACCY}$) and its output connected to NAND gates **328** and **330,** has been replaced by a NAND gate **652,** which has an additional input connected to receive the availability signal $\overline{AVAS}$. Thus, NAND gate **328** will set the call above flip-flop **330** when there is a car or an up corridor call at the floor of the car and the car is available ($\overline{AVAS}$ = 0), or there is a request to accelerate the car ($\overline{ACCY}$ = 0). NAND gate **330** will set the call below flip-flop **332** when there is a car call or a down corridor call at the floor of the car and the car is available, or there is a request to accelerate.

The slowdown signal SLDN is generated by a NAND gate **654** and a flip-flop **656** having cross-coupled NAND gates **658** and **660.** The inputs of NAND gate **654** are connected to receive a signal $\overline{DOR}$, which is a door open request by the call selector **92'**, and to receive a signal $\overline{DOPN}$ which is a door open request by the system processor. The output of NAND gate **654** is connected to an input of NAND gate **658** of flip-flop **656.** An input of NAND gate **660** is connected to receive the signal $\overline{DEC}$, and the output of NAND gate **660** is connected to output terminal SLDN, which provides the slow-down signal. When a request is made to decelerate the car, signal $\overline{DEC}$ goes low and signal SLDN goes high. The slowdown signal SLDN remains high until signal $\overline{DOPN}$ or signal $\overline{DOR}$ goes high and the doors start toclose. If the stop was not responsive to a call and the doors do not open, signal SLDN will terminate when deceleration is completed.

A JK flip-flop **662** has been added to provide a signal NCL at its $\overline{Q}$ output responsive to flip-flop **380.** The NCS output of flip-flop **380** is connected to the J input and the $\overline{NCS}$ output is connected to the K input. The clock input C is connected to receive timing signal **S200.** Thus, when there are no calls and signal NCS goes to logic 0, signal NCL will go to logic one at timing pulse **S200** and remain at the logic one level, at least until the next timing pulse. Signal NCL is used by the system processor **512.**

The serial advanced car position signal EQIR from comparator **82** of FIG. **14** is the A=B output of the comparator and it exists for a slightly longer period of time than the strobed serial advanced car position signal EQIZ.

FIG. **17**

FIG. **17** is a block diagram of the input channel **669** of interface **542** of the system processor **512** shown in FIG. **11.** The serial data from the plurality of elevator cars is individually received over a plurality of high speed data links **LC5.** The letters A, B, C and D identify the various cars sending data. If the data is transmitted from the car controller by light coupled links, blocks **670, 672, 674** and **676** represent receivers for changing the optical data to electrical signals. A light coupled photoemitter-photodiode pair with suitable amplification of the photodiode signal eliminates ground loop noise which might interfere with the signals. If the data link is not light coupled, the receivers would not be required.

The data from the LC5 high speed data links is applied to a car selector **678,** which successively connects its output to different LC5 inputs from the various cars. The output signal MCDA is a serial stream of car data which is periodically assembled in shift register **680** and transferred in parallel to memory **544.** The serial stream of car data is also sent to the master car controller **504** via high speed data link **LC3.** If the data link LC3 is light coupled, block **682** represents a transmitter for driving the photoemitter. If the link is not light coupled, transmitter **682** is not required.

The LC3 and LC5 data links are referred to as being high speed, because of the high data transmission rate, compared with the data rate over the traveling cable, as hereinbefore described relative to the dual data rate time multiplexed concept shown in FIG. **9.**

FIG. **18**

FIG. **18** is a block diagram of the output channel **683** of interface **542** of the system processor **512** shown in FIG. **11.** The data for the various cars is taken in parallel from the memory **544** and loaded into a register **684.** The data in the register **684** is loaded into a multiplexer **686.** The data from the corridor call multiplexer **540** of the master car controller **504** shown in FIG. **11** is received over data link **LC1,** such as via receiver **688** if light coupled, and information from the master car controller, such as corridor calls, is also applied to multiplexer **686.** The output from multiplexer **686** is a serial stream of car data for the various cars of the bank. The data for a specific car appears in a predetermined sequence and this car takes the data off of the serial data line at the appropriate time. Block **689** indicates a malfunction detector which sends the signal $\overline{EMT}$ to the cars when the system processor **512** malfunctions. For example, block **689** may represent a timing circuit which must be accessed by the system processor at precise predetermined times. When it is not accessed properly, signal $\overline{EMT}$ is generated which puts the cars on independent control. The car data output words are sent from the multiplexer **686** to the plurality of car controllers via a plurality of high speed data links **LC8.** If these links are light coupled, the data for cars A, B, C and D, for example, would be sent via transmitters **690, 692, 694** and **696,** respectively.

FIGS. **19–22**

FIGS. **19, 20, 21** and **22** illustrate data link maps for the high speed data links **LC1, LC3, LC5** and **LC8,** respectively, which may be used. The data link maps illustrate the basic scan slots vertically along the left hand side, with FIG. **19** illustrating the binary address of each scan slot as developed by scan counter output **S0S–S6S.** The subdivision of each of the basic scan slots is shown horizontally under the heading "high speed scan." The addresses of the high speed scan slots are developed by clock signals **KO4, KO2, KO1** and **KP5,** all of which are illustrated in FIG. **13.**

For purposes of example, it will be assumed that each of the basic scan slots exits for two milliseconds, which provides a data transmission rate over the traveling cable of 500 hz, a rate which provides an adequate update of car calls and car signals, as well as signal intelligibility over the long, electrically noisy traveling cabele. Each basic scan slot is divided into sixteen bits by the high speed scan, transmitting monitoring and command data between the system processor **512** and the car controllers at a rate of 8 khz, which rate is adequate for efficiently monitoring and providing command signals for bank operation of a group of elevator cars by a system processor. Also, since the distance between the system processor and car controllers in the

penthouse is relatively short, this rate is consistent with signal intelligibility.

Each floor of the building to be served by the elevator system is assigned one of the basic scan slots. The number of floors plus the number of scan slots required to identify express zones, and the like, determines how high the scan counter should be programmed to count before resetting to zeros. As illustrated in FIG. **12, 16, 32, 64** or **128** scan slots may be chosen, each of which will have a predetermined binary address set by the scan counter signals **S0S–S6S**. For purposes of example, it will be assumed that the data link maps are associated with a structure having **26** floors, which includes a basement floor, floors numbered **1** through **24,** and a top extension floor. Thus, the scan counter may be programmed to count from 0 to 31 in binary before resetting, which provides 6 scan slots which may be used for express zone information or other uses. Each of the floors of the structure are assigned a binary address of the scan counter, as illustrated in the right hand vertical column of FIGS. **19–22.** When the scan counter is outputting the address of a specific floor, a car call for that specific floor will appear in that basic scan slot, as illustrated in FIG. **10.** A car signal may also appear during this same basic scan slot, also as illustrated in FIG. **10.** During the same address of the specific floor, the high speed scan will output a plurality of bits of information relative to this same floor, as illustrated in FIG. **10,** and also in FIGS. **19–22.** Thus, when the scan counter output is 01001, the binary address of the eighth floor, data concerning the eighth floor is transmitted over both the low speed and high speed time multiplexed links.

Specifically, FIG. **19** illustrates a data link map which may be used for data link **LC1** which is a data channel from the master car controller **504** to the system processor **512.** Data from a traffic director station may be transmitted during certain of the high speed scan slots, such as slots **0** through **5** and **9** through **14,** one of the slots may be used to check parity, such as slot **15,** and certain of the slots may be used for down corridor calls $\overline{\text{DNC}}$, up corridor calls $\overline{\text{UPC}}$, and a floor enable signal FEN, such as slots **6, 7** and **8,** respectively. Thus, when the basic scan slot **9** exists (address 01001, which has been assigned to the eighth floor), a down corridor call $\overline{\text{DNC}}$ for the eighth floor will appear in the sixth high speed scan slot, and an up corridor call $\overline{\text{UPC}}$ for the eighth floor would appear in the seventh high speed scan slot, and the floor enable signal FEN for the eighth floor would appear in the eighth high speed scan slot. Special calls, such as those from the top extension and basement may appear in yet another high speed scan slot, such as slot **9,** during the appropriate basic scan slot.

FIG. **20** illustrates a data link map which may be used for high speed link **LC3,** which is a data channel from the system processor **512** to the master car controller **504.** The primary purpose of this link is to send the corridor call resets to the master car controller, with high speed scan slots **6, 7** and **8** being illustrated as assigned to down corridor call reset signal $\overline{\text{DNRZ}}$, up corridor call reset signal $\overline{\text{UPRZ}}$, and special call resets, respectively. In addition, the traffic director station, which is not illustrated since it is not necessary to understanding the invention, may receive the input data words sent from the cars to the system processor. Three input data words from each car are illustrated as **IW0, IWI,** and **IW2,** with the car identified by the letter C and the letter assigned to the car. For example, the first input word from car A is identified with the letters **IW0–CA.**

The input words are assigned specific scan slots of the specific scan cycle. For example, the first input word **IW0** for cars A, B, C, and D may occupy basic scan slots **4, 5, 6** and **7,** respectively, the second input word **IW1** for cars A, B, C and D may occupy basic scan slots **8, 9, 10** and **11,** respectively, and the third input word for cars A, B, C and D may occupy basic scan slots **12, 13, 14** and **15,** respectively. Basic scan slots **0–3** may be used to provide system signal data, such as up peak, down peak, demand for top extension, demand for basement service, and the like. The format llsted above is then repeated after every sixteen basic scan slots.

By way of example, the first input word **IW0** may include signals SLDN (car slowing down), BYPS (car bypassing), $\overline{\text{INSC}}$ (car in service with system processor), UPTR (up travel), UPSV (up service), CALL (actual car call), CCAB (car call above the advanced car position), CCBL (car call below the advanced car position), DRCL (doors closed), $\overline{\text{32L}}$ (car in motion), and AVAS (car available for assignment). Signal $\overline{\text{32L}}$ may be provided by a relay responsive to car motion, and the generation of the signal is thus not shown. The high speed scan slots allocated to these signals are illustrated in FIG. **20.** The second input word **IW1** may include the advanced car position in binary signals AVP0 through AVP**6,** as illustrated in FIG. **20.** The third input word **IW2** may include signals ATVS (car on attendant service), CREG (car call registered), $\overline{\text{WT50}}$ (car load greater than **50%** of capacity), and $\overline{\text{WT75}}$ (car load greater than **75%** of capacity). The signal ATVS may be provided in response to a switch (not shown) in the car. The second incorporated application illustrates how the input words are utilized by the system processor to make decisions, assignments, and the like.

FIG. **21** illustrates a data link map which may be used for high speed link **LC5** which is a data channel from each car controller to the system processor **512.** The down corridor reset signal $\overline{\text{DNRZ}}$, up corridor reset $\overline{\text{UPRZ}}$ and car calls $\overline{\text{CCY}}$ may occupy high speed scan slots **6, 7** and **8,** respectively, the parity check may be made in high speed scan slot **15,** while the remaining high speed scan slots may contain the car status signals from each car controller, as hereinbefore described relative to data link **LC3** including the input data words **IW0, IW1,** and **IW2,** as well as system signal words.

FIG. **22** illustrates a data link map which may be used for high speed link **LC8,** which is a data channel from the system processor **512** to each car controller. The down corridor calls $\overline{\text{DNC}}$, up corridor calls $\overline{\text{UPC}}$ and floor enable signal FEN for the various floors may appear in high speed scan slots **6, 7,** and **8,** respectively.

Basic scan slots **0** through **3** may include traffic director station information, as well as the signal $\overline{\text{EMT}}$ which places the cars on independent control when true, free of a system processor.

Basic scan slots **4** through **15** may include the status and command signals from the system processor, which may be grouped into three output words called OWO, OW1 and OW2. The output word for each car is identified with the letter C and the car letter. For example,

the first output word for car A is identified as OWO–CA.

The first output word OWO may include such signals as $\overline{PARK}$ (parking command for the addressed floor), MODO and MOD1 (floor address mode signals), TASS (travel assignment), SASS (service assignment), and the floor address signals FAD0 through FAD6.

The second output word OW1 may include such signals as $\overline{BSMT}$ (basement assignment), $\overline{MCCR}$ (master call reset), $\overline{CCAI}$ (car call answer inhibit), $\overline{DOPN}$ (open door), $\overline{DCLO}$ (close door) and HLM0 and HLM1 (hall lantern mode signals).

The third output word OW2 may include such signals as $\overline{NEXT}$ (car next), $\overline{MNFL}$ (advanced car position at main floor), and $\overline{STT}$ (special service).

The second incorporated application illustrates how the output words are developed.

The high speed data links described will handle up to and including 4 cars. When more than four cars are in a bank, additional data links would be run in parallel with those described in order to carry the additional car information.

FIG. 23

FIG. 23 is a core map which gives, by way of example, the location of the corridor and car calls, as well as the input/output words of the system processor, in the memory 544 of the system processor 512.

The word name is listed in the first vertical column, the address of this word is listed next, and 12 bits of information relative to this word are then listed.

The first 128 addresses, only 6 of which are shown, are for car and corridor calls. The car and corridor word CL is followed by the basic scan slot number of each word. Car calls for up to 8 cars are arranged in bits 0–7 of each word, and down and up corridor calls are arranged in bits 8 and 9, respectively. If the car has a rear door, down and up corridor calls for the rear door may be arranged in bits 10 and 11, respectively. Thus, a car call in car A for scan slot 2, which may be floor number 2 as illustrated in the figures of the data links, would appear as signal CC002 in bit 0 of core address 0000010.

The car signals for car A, IW0, IW1, IW2, OW0, OW1 and OW2 appear at the addresses listed in FIG. 23, with the bit location of the word in the core being as illustrated. The signals for the remaining cars are then listed.

The corridor and car call words, as well as the car signal words may be placed in the memory 544, or retrieved therefrom, via a direct memory access channel between each car controller and the system processor.

FIG. 24

FIG. 24 is a block diagram which illustrates the functions of multiplexer and control 536 and the multiplexer and assignment logic 554 of each car controller, as illustrated in FIG. 11. Multiplexer in control 536 includes means 700 for receiving data link LC8 from the interface 542 of the system processor 512. Means 700 may simply be a receiver when link LC8 is light coupled, for changing the serial optical data to serial electrical data TRI. In like manner, control 536 includes means 702 and 704 for receiving the timing and synchronization data LCC and LCS, respectively, which originated in the master timer 516, and generating timing and synchronization signals SC and SS, respectively, for the slave timing function 514. When the timing and synchronization links are light coupled, means 702 and 704 are receivers for changing the serial optical data to serial electrical data.

A plurality of mutliplexers, such as multiplexers 706, 708 and 710 are provided for assembling parallel car data input and multiplexing the car data serially to the interface 542 of the system processor via high speed data link LC5. Down and up corridor reset signals $\overline{DNRZ}$ and $\overline{UPRZ}$ from the floor selector are applied to each multiplexer 706, 708 and 710, as are the car calls $\overline{CCY}$. These signals appear in high speed scan slots 6, 7 and 8, respectively, of data link LC5, as illustrated in FIG. 21. The parallel data bits from the floor selector for providing input words IW0, IW1 and IW2 are applied to multiplexers 706, 708 and 710, respectively, with the specific signals which may be in these words being hereinbefore described relative to FIG. 20. The signals CALL and CREG which are included in input words IW0 and IW2, respectively, are developed by a circuit 712 using the car call signal $\overline{CCY}$ and the car signal $\overline{CLPB}$.

FIG. 25

FIG. 25 is a circuit which may be used for the block 712 of the control 536. Circuit 712 shown in FIG. 25 includes a terminal connected to receive car calls $\overline{CCY}$, terminals connected to receive timing signals S4, S300 and S200, a terminal connected to receive a signal $\overline{CLPB}$ which is true when a car call push button is depressed. Circuit 712 includes NOT gates or inverters 714, 716, 718 and 720, flip-flops 721, 723 and 725, and NAND gates 722 and 732. Flip-flop 721 may be of the cross-coupled NAND gate type, having NAND gates 724 and 726. Flip-flop 723 may also be of the cross-coupled NAND gate type having NAND gates 728 and 730. Flip-flop 725 may be of the D-type which transfers input information to the output on the positive edge of the clock pulse.

Input terminal $\overline{CCY}$ is connected to an input of NAND gate 722 via inverter 714, and input terminal S4 is connected to the other input of NAND gate 722. Thus, the output of NAND gate 722 will be driven low when an actual car call is found during the scan. The output of NAND gate 722 is connected to an input of NAND gate 724 of flip-flop 721. Thus, when an actual car call is found, flip-flop 721 is set, driving the output of NAND gate 724 high. The output of NAND gate 724 is connected to the D input of flip-flop 725, and the high D input drives the $\overline{Q}$ output low on the leading edge of clock pulse S200. The low output of $\overline{Q}$ is inverted to a true CREG signal by inverter 720. The CREG signal remains true for at least one complete scan cycle, between two successive S200 signals which appear only in scan slot 000, even though flip-flop 721 is reset by clock pulse S300 via inverter 716.

The output signal CALL is developed either by the $\overline{Q}$ output of flip-flop 725 going low, or by a true signal $\overline{CLPB}$, which ever occurs first. Input terminal $\overline{CLPB}$ is connected to an input of NAND gate 728 of flip-flop 723, and when signal $\overline{CLPB}$ goes low, flip-flop 723 is set to provide a logic one output from NAND gate 728. This logic one is inverted by inverter 718 to a logic zero, and applied to NAND gate 732, which outputs a true signal CALL. The Q output of flip-flop 725 is also connected to an input of NAND gate 732, driving signal CALL true at clock pulse S200 after a call is picked up during a scan. A low Q output resets flip-flop 723. The $\overline{CLPB}$ car signal is multiplexed up from the car sta-

tion to the car controller more frequently than individual car calls, and thus time may be saved by using the $\overline{CLPB}$ signal to determine if a car call has been registered.

Referring again to FIG. **24**, each of the multiplexers **706**, **708** an **710** are enabled for a quadrant of four basic floor scans via signals from timing control **742**. If additional data is provided for the first four basic scan slots, such as data for the traffic director station, it would be loaded into still another multiplexer **740** and enabled via signals from the timing control **742** for the first four scan slots. Multiplexers **706**, **708** and **710** would then be successively enabled for scan slots **4–7**, **8–11** and **12–15**, respectively, as illustrated in FIG. **21**.

The demultiplexer and assignment logic functions illustrated by block **554** of FIG. **11** are illustrated in FIG. **24**. The serial data from the dispatcher, now appearing from control **536** as signal TRI is assembled by clocking it into a **16** bit shift register **750** via clock signal FSTA. At the end of each **16** scan slots a complete word has been assembled in shift register **750**, and **12** bits of this word are transferred into a selected one of several storage registers **752**, **754**, **756** or **758**. The selection is made by timing control **742** of control **536**, via signals LW0, LW1, LW2 and LW3 for registers **752**, **754**, **756** and **758**, respectively, providing the three output words and traffic director station information, respectively. The corridor call and floor enable information is transferred to the call register **760**, which provides the up corridor calls 1Z, down corridor calls 2Z, and floor enable signal FEN.

FIG. **26**

FIG. **26** is a schematic diagram of a call register which may be used for the block function **760** shown in FIG. **24**. When signal FSTA has clocked in **8** bits of information to shift register **750** of FIG. **24**, down corridor calls $\overline{DNC}$, up corridor calls $\overline{UPC}$ and floor enable signal FEN appear in the initial three bits of the shift register **750**, and may now be removed for processing. This may be accomplished by strobing the first three bits of shift register **750** immediately following the ninth FSTA signal of a scan. This occurs during high speed scan address HA08, as shown in FIG. **13**. A three bit call register **762** is connected to the initial three bits of shift register **750**, via input terminals **766**, **768** and **770**, and its read enable line **772** is connected to clock signals FSTB and HA08 through AND gate **764**. When signals HA08 and FSTB coincide, the information on input terminals **766**, **768** and **770** is transferred to its output terminals **774**, **776**, and **778**. The information remains at its output terminals through 7 high speed scanning address signals, until its reset line **780** is driven low by clock $\overline{FST0}$ during scanning address HA00.

Outputs **774** and **776**, which carry down and up corridor calls, respectively, are connected to inputs of NAND gates **782** and **784**, respectively. Output **778** is connected to an inverter **786** which provides the floor enable signal FEN. The floor enable signal FEN, the corridor call enable signal CREN, and the bypass signal $\overline{BYPS}$ are all connected to inputs of a NAND gate **788**. The output of NAND gate **788** is connected to inputs of NAND gates **782** and **784** via an inverter **790**. When a scan slot is enabled by a true or high floor enable signal FEN, and by a high corridor call enable signal CREN, and the car is not bypassing ($\overline{BYPS}$ = 1), NAND gate **788** enables NAND gates **782** and **784** via inverter **790** for that specific scan slot. A down corridor call for this scan slot then drives the output of NAND gate **782** low, providing a true down call signal $\overline{2Z}$. An up corridor call for this scan slot drives the output of NAND gate **784** low, providing a true up call signal $\overline{1Z}$.

Returning again to FIG. **24**, the system processor **512** controls the operation of a bank of elevator cars by serial inhibits, which, when the system processor fails to operate, automatically places the cars on independent control, as the system calls are not inhibited but are allowed to be considered by all of the cars. The interfacing of the programmable system processor with the selectors of a group of elevator cars has been hereinbefore described relative to the block diagrams of FIGS. **4** and **5**. The elevator system **500** shown in FIG. **11** utilizes the concept of FIG. **5**, i.e., the system processor is not limited to merely showing a car a single floor at a time. In implementing this concept, as will now be described, the concept of FIG. **4** (single address interfacing) is also described.

More specifically, FIG. **24** illustrates call inhibit logic means **792**, which provides corridor call inhibit signal $\overline{CIN}$ in response to the output signals SOS-S6S of the scan counter, the floor address signals FAD0-FAD6, and the address mode signals MOD0 and MOD1. The floor address and mode signals are provided by the system processor **512** in response to programmed strategy, and appear in the first output word OW0.

FIG. **27**

FIG. **27** is a schematic diagram of a call inhibit logic cirucit which may be used for the call inhibit logic **792** shown in FIG. **24**. An assignment register or comparator **800** is provided which compares the binary floor address FAD0–FAD6 with the binary outputs S0S–S6S of the scan counter. The floor address provided by the system processor **512** will be referred to as word A, while the output of the scan counter will be referred to as word B. With the single address interfacing concept shown in FIG. **4**, the A=B output would be driven high when the scan counter outputs the binary address of the selected floor, which output would be used to generate the inhibit signal $\overline{CIN}$. With the "family of call" interfacing concept shown in FIG. **5**, however, five NAND gates **802**, **804**, **806**, **808** and **810** and two inverters **812** and **814** are added to the inhibit processing circuitry, and the A>B and A<B outputs of comparator **800** are used, in addition to the A=B output. When the scan counter is scanning floor addresses located below the selected floor (FADO–FAD6), output A>B is high or true, while output A<B is true when the scan counter is scanning floor addresses located aboove the selected floor. When the scan counter output word B and the selected word A are equal, the A = B output goes high, which output is connected to an output terminal PFL. The PFL signal appearing at this terminal is used along with the signal $\overline{PARK}$ from the system processor, to provide the forced or dummy car call signal $\overline{FCC}$, as will be hereinafter explained.

In addition to the floor address inputs FAD0–FAD6 and scan counter inputs SOS–S6S, call inhibit logic **792** has inputs connected to receive the address mode signals MOD0 and MOD1, as well as the in service signal INSC and the through trip signal $\overline{EMT}$.

NAND gate **802** has inputs connected to the A<B output of comparator **800**, address mode bit MOD1, and address mode bit MOD0 via inverter **812**.

NAND gate **804** has inputs connected to the A>B output of comarator **800**, and to address mode bit signals MOD0 and MOD1.

NAND gate **806** has inputs connected to the address mode bits MOD0 and MOD1 via inverters **812** and **814**, respectively.

NAND gate **808** has inputs connected to the A = B output of comparator **800** and to the output of NAND gate **806**. The outputs of NAND gates **802**, **804** and **808**, as well as signals INSC and $\overline{EMT}$, are all connected to inputs of NAND gate **810**. The output of NAND gate **810** provides the corridor call inhibit signal $\overline{CIN}$.

The floor address mode bits select one of the inhibit modes shown in FIG. **6**. If both mode bits are at the logic zero level, the car is in service with the system processor (INSC = 1), and the system processor is functioning normally ($\overline{EMT}$ = 1), the car cannot consider any corridor calls, regardless of the output signal FAD0–FAD6. When both address mode bits are at the logic zero level, NAND gate **806** outputs a zero, switching NAND gate **808** to the logic one level. Thus, the A = B output of comparator **800** is ineffective. Address mode bit MOD1 causes NAND gate **802** to output a one, rendering the A<B output of comparator **800** ineffective, and both address mode bits MOD0 and MOD1 block gate **804**, causing it to output a one, rendering the A>B output of comparator **800** ineffective. Thus, all inputs to NAND gate **810** are "one's" for any output condition of comparator **800**, which maintains a true inhibit signal $\overline{CIN}$ for all floors.

When the system processor **512** outputs a logic one for address mode bit MOD0 and a logic zero for address mode bit MOD1, the inhibit signal $\overline{CIN}$ goes high or false only during the time the A=B output is high, enabling the elevator car to consider only calls at the addressed floor. The signal UPSV determines whether an up or down call at the addressed floor will be considered. This corresponds to the single address interfacing concept of FIG. **4**. The address mode bit MOD1 blocks NAND gates **802** and **804** into their high output conditions, rendering the A<B and A>B outputs of comparator **800** ineffective. NAND gate **806** outputs a logic one due to the inverted MOD0 signal, enabling NAND gate **808**. When the A=B input to NAND gate **808** goes high, NAND gate **810** outputs a false or high $\overline{CIN}$ signal during this time slot, removing the inhibit for the floor of this selected scan slot.

When the system processor **512** outputs the address mode bits MOD0 and MOD1 at the logic zero and logic one levels, respectively, the car may consider corridor calls at the addressed or selected floor and for all of the floors above the selected floor. NAND gate **804** is blocked by the low MOD0 signal, rendering the A>B output ineffective. NAND gate **808**, however, is enabled, driving the output signal $\overline{CIN}$ high when the A=B output is true, and NAND gate **802** is enabled, driving the output signal $\overline{CIN}$ high when the A<B output is true. Thus, the only calls blocked out or inhibited are those from the floors located below the FAD0–FAD6 address. Signal UPSV determines whether the car will consider up or down corridor calls at the enabled floors.

When the system processor **512** outputs both address mode signals MOD0 and MOD1 at the logic one level, the elevator car may consider corridor calls at the selected floor and all of the floors below the selected floor. The inverted MOD0 signal blocks NAND gate **802**, rendering the A<B output of comparator **800** ineffective, while NAND gate **806** enables NAND gate **808**, and NAND gate **804** is enabled. When the A=B output of comparator **800** is true, NAND gate **810** outputs a high $\overline{CIN}$ signal, and when the A>B output is true, NAND gate **810** outputs a high $\overline{CIN}$ signal. Thus, the only calls blocked or inhibited are those from the floors located above the FAD0–FAD6 address. Again, signal UPSV determines whether the car will consider up or down calls at the enabled floors.

If a car is not in the service of the system processor **512**, the signal INSC goes low, forcing the output of NAND gate **810** to provide a high $\overline{CIN}$ signal. If the system processor **512** malfunctions, and the circuit which provides signal $\overline{EMT}$ is not accessed properly by the system processor **512**, signal $\overline{EMT}$ goes low which also forces a high $\overline{CIN}$ signal, removing the car from inhibit control.

FIG. **28**

FIG. **28** is a forced or dummy car call circuit **816** which illustrates how the signal PFL provided by the call inhibit logic means **792**, along with the signal $\overline{PARK}$ from the system processor (bit 0 of word OW0), are used to provide the forced car call signal $\overline{FCC}$ hereinbefore described relative to FIG. **15**. In addition to signals PFL and $\overline{PARK}$, circuit **816** uses signal $\overline{DL2}$, which is a logic zero when the car is to run, and a logic one when the car is to land, signal AVAS which is true when the car is available to accept an assignment, signal EQIZ, which is true in the scan slot corresponding to the serial advanced car position, and timing signals S4 and STA shown in FIG. **13**.

Circuit **816** utilizes NAND gates **818**, **820**, **822** and **824**, a flip-flop **826** which may have cross-coupled NAND gates **828** and **830**, and an inverter **832**. Input signals PFL, EQIZ and $\overline{DL2}$ are connected to inputs of NAND gate **818**, and the output of NAND gate **818** is connected to an input of NAND gate **828** of flip-flop **826**, which may be called the parking flip-flop. NAND gate **828** has another input connected to receive signal $\overline{PARK}$ via inverter **832**. NAND gate **820** has inputs connected to receive signals EQIZ and $\overline{DL2}$, and its output is connected to an input of NAND gate **822**. The remaining inputs of NAND gate **822** are connected to receive signals PFL, PARK, AVAS and S4. The output of NAND gate **822** is connected to an input of NAND gate **830** of flip-flop **826**. The output of NAND gate **830** is connected to an input of NAND gate **824**, and the remaining input of NAND gate **824** is connected to receive strobe STA. The output of NAND gate **824** provides the forced car call signal $\overline{FCC}$, which parks the car at the addressed floor without operating the hall lanterns or opening its doors, as hereinbefore described relative to FIG. **15**.

When signal $\overline{PARK}$ is zero, and the car is available, (AVAS = 1), NAND gate **822** will output a zero at time S4, setting the parking flip-flop **826** to provide a logic one input to NAND gate **824**. At time STA, the output of NAND gate **824** goes low to provide a ture $\overline{FCC}$ signal. As hereinbefore explained, the forced or dummy call $\overline{FCC}$ appears during time slot STA to differentiate it from a real car call, which appears during time S4.

If a car is already at the specified floor, the inputs to NAND gate **818** will all be high providing a logic zero input to NAND gate **828** and the output of NAND gate **820** will be low, forcing NAND gate **822** to output a logic one. Thus, the parking flip-flop **826** will not be set.

When the car arrives at the specified floor, the output of NAND gate **818** will go low, and the output of NAND gate **822** will go high, resetting the parking flip-flop **826.**

Returning to FIG. **24,** control **554** also generates the assignment signals relating to service and travel direction. Signals $\overline{SUS}$ and $\overline{SDS}$, when true, request that the floor selector be set for up service and down service, respectively, and signals $\overline{SUT}$ and $\overline{SDT}$ when true, request that the selector be set for up travel, and down travel, respectively. These assignment signals are generated by assignment logic means **834** in response to the service and travel assignment signals SASS and TASS, respectively, which are part of output word **OW0** from the system processor **512,** which appears in storage register **752.** In addition to these signals, the assignment logic means **834** utilizes signal $\overline{STT}$ provided by the system processor **512** during certain basement and top extension assignments, which appears in output word **OW2,** as well as certain signals from the floor selector, such as the call above and call below signals CA and CB, respectively, the car call above and car call below signals CCAB and CCBL, respectively, the availability signal AVAS, the door open request signal $\overline{DOR}$, the in service signal INSC, the service processor malfunction signal $\overline{EMT}$, and timing signal **S200.**

FIG. **29**

FIG. **29** is a schematic diagram of assignment logic which may be used for the block assignment logic function **834** shown in FIG. **24.** The assignment logic **834** includes a plurality of NAND gates **836, 838, 840, 842, 844, 846, 848, 850, 852** and **854,** and a plurality of inverter gates **856, 858, 860, 862, 864, 866, 868** and **870.**

NAND gate **836** has an input connected to receive the call above signal CA via inverter gate **856,** and another input is connected to receive the car availability signal AVAS via inverter **858.** The output of NAND gate **836** is connected to an input of NAND gate **848.** NAND gate **838** has an input connected to receive the availability signal AVAS via inverter **858,** and an input connected to receive the call below signal CB via inverter **860.** The output of NAND gate **838** is connected to an input of NAND gate **850.** NAND gate **840** has an input connected to receive the car call below signal CCBL, and an input connected to receive the up travel signal UPTR via inverter **862.** The output of NAND gate **840** is connected to an input of NAND gate **852.** NAND gate **842** has an input connected to receive the up travel signal UPTR, and the car call above signal CCAB. The output of NAND gate **842** is connected to an input of NAND gate **854.** NAND gate **844** has inputs connected to receive signals AVAS and $\overline{STT}$ via inverters **858** and **864,** respectively, and the output of NAND gate **844** is connected to an input of NAND gate **846.** THe remaining inputs of NAND gate **846** are connected to receive signals $\overline{EMT}$, INSC, and $\overline{DOR}$. The output of NAND gate **846** is connected via inverter **866** to inputs of NAND gates **848, 850, 852** and **854.**

The travel assignment signal TASS from the system processor **512** is connected directly to an input of NAND gate **848,** and via inverter **868** to an input of NAND gate **850.** The outputs of NAND gates **848** and **850** provide the up travel and down travel assignment signals $\overline{SUT}$ and $\overline{SDT}$, respectively.

The service direction assignment signal SASS from the system processor **512** is connected directly to an input of NAND gate **852,** and via inverter **870** to an input of NAND gate **854.** The outputs of NAND gates **852** and **854** provide the up service and down service assignment signals $\overline{SUS}$ and $\overline{SDS}$, respectively.

The system processor **512** sends travel and service assignment signals TASS and SASS, and the assignment logic **834** determines whether or not to respond to the assignments. The system processor **512** does not monitor the operation of the floor selector close enough to insure that its assignments will always be valid. Thus, the assignment logic **834** determines when the assignments from the system processor are valid, to prevent the system processor from interfering with the selector operation at other times. The system processor **512** may not control the floor selector when the car is not in service (INSC = 0), if the car is on through trip ($\overline{EMT}$ = 0), if the car is on special assignment ($\overline{STT}$ = 0), or during the non-interference time ($\overline{DOR}$ = 0). Further, if there is a car call ahead of the travel direction, the system processor is prevented from setting the opposite service direction.

Specifically, if there is no call above or below the advanced car position (CA and CB = 0), no car call above or below (CCAB and CCBL = 0), the car is available (AVAS = 1), the car is in service with the system processor (INSC = 1), the system processor is functioning properly, ($\overline{EMT}$ = 1) and there is no door open request ($\overline{DOR}$ = 1), the travel and service direction will be under control of the system processor **512.** When the travel assignment signal TASS is at the logic one level, NAND gate **848** will output a low or true set selector for up travel signal $\overline{SUT}$ at time **S200,** while NAND gate **850** will output a high set selector for down travel signal $\overline{SDT}$. A low signal TASS will provide a high signal $\overline{SUT}$, and a low or true signal $\overline{SDT}$ during time **S200.**

A high service assignment signal SASS will provide a true set selector for up service signal $\overline{SUS}$ during time **S200,** and a false set selector for down travel signal $\overline{SDS}$. A low signal SASS, on the other hand, provides a false $\overline{SUS}$ signal, and a true $\overline{SDS}$ signal during time **S200.**

If a car is available (AVAS = 1), NAND gates **836** and **838** provide high outputs, enabling the system processor to change the travel direction at will. If the car is not available (AVAS = 0), there must be a call in the selected direction (CA or CB = 1) before the system processor can set the travel direction. For example, if a car is not available (AVAS = 0) and there is a call above, (CA = 1) NAND gate **836** enables NAND gate **848,** allowing the system processor to set the travel direction for up. If the car is not available and there is a call below, (CB = 1) NAND gate **838** enables NAND gate **850** allowing the system processor to set the travel direction to down. If the car is not available, and there is both a call above (CA = 1) and a call below (CB = 1), both NAND gates **836** and **838** will output logic one's enabling the system processor to set the travel direction for up or down.

If there are no car calls, signals CCBL and CCAB will be at the logic zero level, and the system processor may change the service assignment at will. When the car has a call reversal assignment, i.e., a service assignment opposite to the travel assignment, and a car call is entered ahead of the car in the travel direction due to a late car call registration by a passenger, the selector will change the service direction to agree with the travel direction, and the system processor must not be permitted to interfere with this change. Therefore, a car call below (CCBL = 1) when the car is traveling down (UPTR = 0) provides a zero output from NAND gate **840** which prevents the service assignment signal SASS from providing a true $\overline{SUS}$ signal. A car call above (CCAB = 1) when the car is traveling up (UPTR = 1) provides a zero output from NAND gate **842,** preventing the service assignment SASS from providing a true $\overline{SDS}$ signal.

When the car is on certain basement or top extension assignments, the system processor **512** issues a true signal $\overline{STT}$ which places the car on a special type of through trip, similar to when the car is removed from service from the system processor and placed on through trip by a true $\overline{EMT}$ signal. This approach to basement and top extension service relieves the system processor from monitoring the car while on the special assignments, conserving core space in the memory **544.**

When signal $\overline{STT}$ is true and the car is not available (AVAS = 0), NAND gate **844** outputs a logic zero which forces NAND gate **846** to output a logic one, which is inverted by inverter **866** to the logic zero level, blocking NAND gates **848, 850, 852** and **854** in their high output states. Thus, the floor selector receives no travel or service assignments from the assignment logic **834** during this period. As hereinbefore described relative to FIG. **16B**, a true $\overline{STT}$ signal drives both the UPSV and DNSV signals high, thus effectively operating the floor selector without service direction control during this period. The system processor **512** notes when the car returns from the basement or top extension service, and the special through trip signal $\overline{STT}$ is driven high to remove the car from the special service and return it to control by the system processor. It will be noted that the through trip signal $\overline{EMT}$ when true, or the in service signal INSC when false, or the door open request signal $\overline{DOR}$ when true, will also drive the output of NAND gate **846** high, blocking the assignment logic **834** from providing any true travel or service assignments.

FIG. **30A**

FIG. **30A** is a block diagram which illustrates the car and penthouse signal control **530** and the car call and car signal control **526** of FIG. **11,** in greater detail. Car calls $\overline{CCLZ}$ from the main push button station **520,** and car calls $\overline{ACALLS}$ from the auxiliary car station control **524,** along with car signals, are time multiplexed by a multiplexer **900** to provide a serial signal $\overline{READ}$. The multiplexed signal $\overline{READ}$ includes in each scan slot, a car call for this scan slot in the first half of the signal, and a car signal in the last half of the scan slot, as illustrated in FIG. **13.** Car signals, for example, include signal $\overline{CLPB}$ which, when true, indicates that a car call push button has been actuated, car direction signals $\overline{ATUP}$ and $\overline{ATDN}$ as set by an attendant, and hereinbefore illustrated in FIG. **16A,** and a signal which indicates when an attendant is bypassing corridor calls. The signal $\overline{READ}$ is at the relatively low voltage used in logic circuits when it leaves multiplexer **900,** and it is changed to a high voltage signal $\overline{PREAD}$ by a logic level/high voltage interface **902.** Signal $\overline{PREAD}$ is increased to a high voltage, as hereinbefore explained, to reduce the possibility of signal interference due to electrical noise while the signal is being transmitted over the traveling cable **532** to the car and penthouse signal control **530.** At the control **530,** the signal $\overline{PREAD}$ is reduced to the voltage level used in logic circuits by a logic level/high voltage interface **904,** again providing a signal $\overline{READ}$, which is applied to a partial demultiplexer **906.** Demultiplexer **906** removes the car calls from the car signals, providing serial car calls in a signal $\overline{READ}$ (1) and serial car signals in a signal $\overline{READ}$ (2). The car signals $\overline{READ}$ (2) are demultiplexed in a demultiplexer **908.** The serial car calls $\overline{READ}$ (1) are sent to car call and reset control **534.**

Penthouse signals, such as the signals for driving car direction indicators, attendant gongs, buzzers which warn that the door will close after being held open for a predetermined period of time, and the like, are serialized in a multiplexer **910** to provide serial penthouse signals $\overline{PSZ}$. The serial penthouse signals $\overline{PSZ}$ and the serial advanced car position signal EQIR are time multiplexed into basic scan slots by a multiplexer **912,** providing a multiplexed signal $\overline{WRIT}$ which includes the serial advanced car position signal EQIR in the first half of the time slot, and a penthouse signal in the last half, as shown in FIG. **13.** Signal $\overline{WRIT}$ is increased in voltage in interface **904,** providing a signal $\overline{PWRIT}$ for transmission over the traveling cable. Signal $\overline{PWRIT}$ is reduced in voltage to logic levels in interface **902** of control **526,** and the $\overline{WRIT}$ signals are demultiplexed, stored and displayed in means **914.**

Signals CLOCK, $\overline{SYNC}$, $\overline{CCR}$, and $\overline{CCS}$, corresponding to timing, synchronization, car call reset, and car call set, respectively, are sent from the car controller to the car station over the traveling cable, proceeding through interfaces **904** and **902** to obtain the voltage changes required for transmission via a traveling cable, and for use in the car station.

Signals $\overline{CSET}$ and $\overline{ASET}$ provide the serial car call set signals for the main and auxiliary car stations **520** and **522,** respectively. The car calls $\overline{CCLZ}$ and $\overline{ACALLS}$ and car sets $\overline{CCS}$ are combined and gated by a false or high car call reset signal $\overline{CCR}$ in control **916** to provide the serial car call set signals. However, it would also be suitable for signal $\overline{CSET}$ to be responsive directly to signal $\overline{ACALLS}$, and for signal $\overline{ASET}$ to be directly responsive to signal $\overline{CCLZ}$.

FIG. **30B**

FIG. **30B** is a schematic diagram of controls **916** which may be used for the functions shown in block form in FIG. **30A.** Control **916** includes NAND gates **954, 956** and **958,** NOT gates **960, 964** and **968,** and a D-type positive edge triggered flip-flop **972.** Control **900** includes NAND gate **962,** NOT gates **966** and **970,** and a D-type positive edge triggered flip-flop **974.**

Car calls $\overline{CCLZ}$ and $\overline{ACALLS}$ from the main and auxiliary car stations **520** and **522,** respectively, are connected to inputs of NAND gate **954.** Thus, any car call drives the output of NAND gate **954** high, providing the signal CALLS which includes all car calls, regardless of which car station they were registered on. Signal CALLS is inverted by NOT gate **964** and applied to an input of NAND gate **956.** The other input of

NAND gate **956** is connected to receive car call set signal $\overline{CCS}$, which signal is initiated by maintenance personnel in the penthouse. Thus, any car call, or any car call set from the penthouse will drive the output of NAND gate **956** high. The output of NAND gate **956** is applied to an input of NAND gate **958.** The other input of NAND gate **958** is connected to receive car call reset signal $\overline{CCR}$. If there is no reset for a time slot, a car call or car call set for this time slot is gated through NAND gate **958** and applied to the D input of flip-flop **972.** Strobe STBS gates this low D input to the Q output on the positive edge of the strobe. The low Q output is inverted by NOT gates **968** and **960** to provide true set signals CSET and ASET for the main and auxiliary car stations, respectively.

If the car call reset signal $\overline{CCR}$ is true for a time slot, this reset takes precedence over a car call or call set for the time slot by forcing the output of NAND gate **958** high, providing false or low signals CSET and ASET for the time slot.

Control **900** includes a multiplexer (not shown) for multiplexing the car signals and placing them in the last one-half of the time slots, as shown in FIG. **13B.** Signal CALLS is applied to the D input of flip-flop **974,** and a true signal CALLS is gated to the Q output on the positive edge of a clock signal which places the car calls in the first one-half of the time slots. A true car signal is low, and a true car call is low due to NOT gate **966.** The car signals and car calls are applied to inputs of NAND gate **962.** Thus, a true car call will drive the output of NAND gate **962** high during the first one-half of the scan slot, and a true car signal will drive the output of NAND gate **962** high during the last one-half of the scan slot. NOT gate **970** inverts these signals to provide the composite signal $\overline{READ}$, which is sent to the car controller in the penthouse.

FIG. **31**

FIG. **31** is a schematic diagram of a car call reset circuit which may be used for providing the car call reset signal $\overline{CR}$, as illustrated by block function **538** in FIG. **11.** Car call reset **538** includes NAND gates **920** and **922,** and an inverter or NOT gate **924.** NAND gate **920** has inputs connected to receive the serial advanced car position signal EQIR, and the request to accelerate signal $\overline{ACCX}$, which is low from the time acceleration is requested until deceleration is initiated. When deceleration is initiated, and signal $\overline{ACCX}$ goes high, the output of NAND gate **920** will be driven low during the scan slot of the advanced car position signal EQIR, which drives the output of NAND gate **922** high during this time slot, which is inverted by inverter gate **924** to a low or true car call reset signal $\overline{CR}$, which resets the car call registered for the floor at which the car will stop. The car call reset signal $\overline{CR}$ may also be driven low by a true signal $\overline{MCR}$, which may be generated in response to predetermined traffic conditions, providing true $\overline{MCR}$ signals in the time slots of floors for which car calls are to be suppressed during the predetermined traffic condition.

FIG. **32**

FIG. **32** is a schematic diagram of a car call and reset circuit which may be used for control **534** shown in FIG. **11.** Control **534** includes NAND gates **930** and **932,** NOR gates **934** and **936,** AND gates **938, 940** and **942,** inverter gates **944, 946, 948** and **950,** and a D-type flip-flop **952.** Serial car call signals $\overline{READ}$ (1) from control **530** are applied to an input of NOR gate **934.** The other input to NOR gate **934** is an enable for the gate responsive to the car call reset signal $\overline{CR}$ and enable signal REN. Enable signal REN may be provided by a read only memory track which provides logic one's in the time slots assigned to floor locations. The car call reset signal $\overline{CR}$ is applied to an input of NAND gate **930** via inverters **944** and **946,** and the enable signal REN is applied to another input of NAND gate **930.** If the time slot is enabled by a high signal REN, and there is no car call reset for this time slot ($\overline{CR} = 1$), the output of NAND gate **930** will be low, enabling NOR gate **934.** If there is a car call for this time slot (floor), signal $\overline{READ}$ (1) will be low, and NOR gate **934** will output a logic one to the D input of flip-flop **952.** D-type flip-flop **952** transfers a signal appearing at its D input to its Q output on the positive edge of the clock pulse applied to its clock input C. The clock pulse is a car call strobe signal $\overline{CCST}$, generated by clock signals HA07 and FSTB shown in FIG. **13,** via NAND gate **932.** When signal CCST is on its positive going leg, flip-flop **952** is actuated, driving its Q output high and providing a true car call signal $\overline{3Z}$ via inverter **948.**

The car call reset signal $\overline{CR}$ and enable signal REN are multiplexed into the proper time slots of the scan counter to provide signal $\overline{CCR}$. Signal $\overline{CCR}$ includes the enable signal REN in the first half of the time slot, and the car call reset signal CRSM in the last half, as shown in FIG. **13.** Car call reset signal $\overline{CR}$ is applied to an input of AND gate **938** via inverter **944,** and the other input of AND gate **938** is connected to receive enable signal REN. A true car call reset $\overline{CR}$ for a time slot which corresponds to a floor (REN = 1) gates AND gate **938** to provide a true car call reset signal CRSM. Signal CRSM is applied to an input of AND gate **940,** and the other input of AND gate **940** is connected to receive clock signal KP5 shown in FIG. **13.** Clock KP5 is true during the last half of the scan slot. Thus, a true car call reset signal CRSM will cause AND gate **940** to output a logic **1** during the last half of the associated scan slot.

The enable signal REN is applied to an input of AND gate **942,** and the other input of AND gate **942** is connected to receive clock KP5 via inverter **950.** Thus, clock KP5 enables a time slot during the first half thereof, and a true enable signal REN will cause AND gate **942** to output a logic 1 during the first half of the associated scan slot.

The outputs of AND gates **940** and **942** are applied to inputs of NOR gate **936,** the output of which is the mutiplexed enable/reset signal $\overline{CCR}$. If both inputs to NOR gate **936** are low, signal $\overline{CCR}$ will be false or high during both halves of the scan slot. If either input is high, signifying a true reset signal CRSM, or a true enable signal REN, signal $\overline{CCR}$ will be low or true during the half of the associated scan slot corresponding to which AND gate is high.

FIG. **33**

FIG. **33** is a schematic diagram of a new car call memory and serializer circuit which may be used to provide block function **520** shown in FIG. **11.** Circuit **520** shown in FIG. **33** utilizes the concept set forth in the block diagram of FIG. **7,** wherein the car call memories are located in the car station, with the car calls being time multiplexed to the car controller, and with the car call resets being time multiplexed to the car station from the car controller. Further, in line with the

concept of FIG. 7, the serialization of car calls occurs in the push button circuits themselves, eliminating a large plurality of wires which would otherwise be required if the multiplexing of the car calls and car signals were to be accomplished by connecting the individual car call memories to the car call and car signal multiplexer. Each push button or call circuit has its output connected to a common call output bus, thus connecting the outputs of the call circuits in parallel to the call bus, and the push button or call circuits are accessed in a predetermined sequence to determine if the associated car call push button has been depressed. FIG 33 illustrates a new concept for sequentially accessing a plurality of push button or call circuits in which car call memories are disposed.

More specifically, FIG. 33 includes a plurality of similar memory elements and associated time multiplexing and interface circuits or modules **1000,** one each for each car call push button. Since each of the circuits **1000** are similar, only one is shown in detail in FIG. **33.**

Only five bus wires **1002, 1004, 1006, 1008** and **1010,** are required, corresponding to call output, call set, call reset, clock and sync buses, respectively. Each circuit **1000** includes memory elements **1012** and **1014,** such as flip-flops, each settable from a first condition to a second condition, NAND gates **1016, 1018,** and **1020,** interface circuits **1022** and **1024,** an indicating lamp **1026,** a car call push button **1028,** and a source of unidirectional potential represented by terminal **1030.**

The corresponding memories **1012** of all of the circuits **1000** are connected in a ring counter arrangement, with the output terminal of memory **1012** being connected to the input terminal of the corresponding memory **1012** of the next circuit **1000.** The clock inputs of these ring counter memories are connected to the clock bus **1008,** and the clear or reset inputs of these memories are connected to the sync bus **1010.** NAND gates **1016** and **1018** each have an input connected to the output of its associated memory **1012,** with NAND gate **1016** also having an input connected to the call set bus **1004,** and NAND gate **1018** also having an input connected to the call reset bus **1006.** The outputs of NAND gates **1016** and **1018** are connected to the set and clear inputs of memory element **1014.** NAND gate **1020** has its inputs connected to the output of memory element **1012,** and to the Q output of memory element **1014.** The Q output of memory element **1014** is also connected to the lamp driver interface **1024,** with the output of the lamp driver interface being connected to lamp **1026.** Lamp **1026** is also connected between ground and the source **1030** of unidirectional potential via push button **1028,** and to high voltage to logic level interface **1022.** The output of interface **1022** is connected to the set input of memory element **1014.**

The car call push buttons are all accessed sequentially by shifting a logical one around the ring counter circuit, with the logical one being shifted each time the scan counter is incremented, to synchronize the accessing of the push buttons with the scan counter. When the scan counter is reset to zeros at the end of a scan, the sync bus **1010** provides a pulse which resets all of the outputs of the ring counter memory elements **1012** to zeros. A logical one is then introduced to the input of the memory element **1012** which corresponds to the lowest floor of the building. Each time the scan counter indexes to a scan slot which corresponds to a floor of the building, a clock pulse is generated on the clock bus **1008,** which shifts the logical **1** from the output of memory **1012** of the first push button circuit **1000** to the input of memory **1012** of the next push button circuit **1000.** The enable signal REN may be used along with a signal which occurs each time the scan counter is incremented, to provide clock pulses on the clock bus **1008** only when the scan slot corresponds to a floor location, as determined by a true signal REN. Thus, each push button is accessed in its designated time sequence and scanning position as defined by the binary scan counter which provides the signals S0S–S6S.

When the push button **1028** is depressed, the lamp **1026** is lit and memory element **1014** is set from its first condition to its second condition via high voltage to logic level interface **1022.** The high Q output of the set memory **1014** drives interface **1024** to retain the illumination of the lamp **1026.** The high Q output of memory **1014** also enables NAND gate **1020.** Now, when the previous memory **1012** applies a logical **1** input to memory **1012,** the next clock pulse from bus **1008** will provide a high output from memory **1012** which, if NAND gate **1020** is enabled by a car call, drives the output of NAND gate **1020** low to signify a car call for the floor of this scan slot. The high output of memory element **1012** enables the memory element **1012** of the next circuit **1000,** which is accessed by the next clock pulse, which process continues until all of the car call push button circuits have been accessed.

When the car call is answered, the reset bus **1006** provides a reset pulse in the proper time slot which, along with the high outputs from memory element **1012** when the memory element is accessed, resets call memory **1014** back to its first condition, removing the drive from the lamp **1026,** and removing the enable from NAND gate **1020.**

If a car call is set for a specific floor by providing a pulse on the call set bus in the proper time slot, NAND gate **1016** will set the call memory **1014** when the circuit is accessed, enabling NAND gate **1020** and driving the lamp interface **1024.**

The call sets are normally generated by maintenance personnel in the penthouse who send the call set signal CSET down via the traveling cable, as illustrated in FIG. **30A.**

When the concept set forth in FIG. **8** is implemented, wherein an auxiliary push button station is utilized, instead of connecting the car call buttons of the main and auxiliary station in parallel, which involves a great deal of wiring in the car station, the auxiliary push button station utilizes call storage and serializer circuits similar to circuits **1000.** Parallel operation of the two push button stations is simulated by connecting the serial signal representing the car calls registered in one station to the set line of the other station. Thus, when a car call push button is depressed and its lamp energized in one station, the call is serialized and placed on the set line of the other station, energizing the proper lamp in such a short time following the energizing of the first lamp that they appear to be in parallel.

FIG. **34**

FIG. **34** is a schematic diagram which illustrates a more detailed implementation of the circuit **1000** shown in FIG. **33,** which may be used. Like reference numerals in FIGS. **33** and **34** indicate like components.

Ring counter memory element **1012** in this embodiment is a JK flip-flop, with its $\overline{Q}$ output connected to the input of the memory element of the next circuit **1000**, via an inverter **1032**. Call memory element **1014** is also a JK flip-flop. The push button and lamp interfaces **1022** and **1024** are isolated in this embodiment, with the push button interface **1022** including an NPN-type transistor **1034** having base, collector and emitter electrodes *b*, *c* and *e*, respectively, resistors **1036**, **1038** and **1039**, a voltage regulating diode **1040**, a capacitor **1042**, and a source of indirectional potential represented by terminal **1044**. The collector electrode c of transistor **1034** is connected to the set input S of flip-flop **1014**, and to voltage source **1044** via resistor **1039**. Its emitter electrode *e* is connected to ground. Its base electrode b is connected to ground via resistor **1036**, and to voltage source **1030** via voltage regulating diode **1040**, resistor **1038**, and push button **1028**. The cathode electrode of voltage regulating diode **1040** is connected to resistor **1038**, and this junction is connected to ground via capacitor **1042**.

The lamp driver interface **1024** includes a PNP-type transistor **1050**, resistors **1052**, **1054**, **1056** and **1058**, diodes **1060** and **1062**, a capacitor **1064**, a solid state switching device **1066**, such as a thyristor having anode, cathode and gate electrodes *a*, *c* and *g*, respectively, and a source of unidirectional potential represented by terminal **1068**.

A source of unidirectional potential **1070** is provided which is a positive half wave source, with preferably a slight negative excursion. The lamp **1026** is connected from voltage source **1070** to ground via thyristor **1066**. Firing pulses for the gate electrode *g* of thyristor **1066** are provided by the remaining portion of the interface, in which the base electrode *b* of transistor **1050** is connected to the $\overline{Q}$ output of flip-flop **1014** via resistor **1052** and diode **1060**. The cathode electrode *c* of diode **1060** is connected to the Q output of flip-flop **1014**. The emitter electrode *e* is connected directly to voltage source **1068**, and via resistor **1054** to its base electrode *b*. The collector electrode c is connected to the gate electrode *g* of thyristor **1066** via diode **1062** and resistor **1056**, with the anode electrode a of diode **1062** connected to the collector electrode *c*. Resistor **1058** and capacitor **1064** are each connected from the gate electrode *g* to ground.

When push button **1028** is depressed, capacitor **1042** charges through resistor **1038**. When the voltage of capacitor **1042** reaches the break down voltage of diode **1040**, transistor **1034** saturates setting the call into flip-flop **1014**, enabling NAND gate **1020**. When $\overline{Q}$ of flip-flop **1014** goes to zero, transistor **1050** saturates, firing thyristor **1066** to illuminate the lamp **1026**. The low $\overline{Q}$ output provides continuous gate drive to thyristor **1066**, refiring it during each half cycle of the source voltage. When the logic one being propagated around the ring counter reaches flip-flop **1012**, and flip-flop **1012** is clocked to provide a high Q output, NAND gate **1020** outputs a call to bus **1002** to provide a car call $\overline{CCLZ}$ in the proper time slot. When the call is reset, signal CCR goes high and NAND gate **1018** provides a logic zero to input terminal R of flip-flop **1014**, resetting this flip-flop. This removes the enable from NAND gate **1020**, and removes the gate drive from thyristor **1066**, which reverts to its non-conductive state at the end of the half cycle of voltage source **1070**.

FIG. 35

FIG. **35** is a schematic diagram of a call storage and serializer circuit **1080** which uses a concept of accessing the various push button circuits which differs from the ring counter accessing concept set forth in FIGS. **33** and **34**. In the concept of FIG. **35**, each push button circuit **1080** includes a seven bit comparator **1082**, with each comparator having a different seven bit binary push button address connected to one set of inputs of the comparator. The other set of inputs of comparator **1082** is connected to a seven bit address bus **1083**, which bus is connected to the comparator of each push button circuit. The address bus is driven by a binary counter which is actuated each time the scan counter advances to a time slot which corresponds to a floor location. Thus, the address bus sequentially accesses all of the push button circuits, providing a pulse from the output of the comparator **1082** when the address on the address bus is the same as the push button address.

Circuit **1080** includes NAND gates **1084**, **1086**, and **1088**, call memory element **1090**, such as a flip-flop, push button and lamp interface circuits **1092** and **1094**, a lamp **1096**, a push button **1098**, and a source of unidirectional potential indicated by terminal **1099**. In addition to the seven bit address bus **1083**, a call output bus **1085**, a call set bus **1087**, and a call reset bus **1089** are provided. NAND gate **1084** has inputs connected to the equality output of comparator **1082** and to the call set bus **1087**. The output of NAND gate **1084** is connected to the set input of flip-flop **1090**. NAND gate **1086** has its inputs connected to the equality output of comparator **1082** and to the reset bus **1089**. The output of NAND gate **1086** is connected to the clear or reset input CLR of flip-flop **1090**. The push button **1098** and lamp **1096** are serially connected from voltage source **1099** to ground. The junction between the push button **1098** and lamp **1096** is connected to input and output interface circuits **1092** and **1094**, respectively. The output of interface **1092** is connected to the set input of flip-flop **1090**. The input of interface **1094** is connected to the output of flip-flop **1090**. NAND gate **1088** has its inputs connected to the equality output of comparator **1082** and to the output of call memory **1090**. The output of NAND gate **1088** is connected to call output bus **1085**.

When push button **1098** is depressed, lamp **1096** is illuminated and call memory **1090** is set via a zero output from push button interface **1092**, which functions to change the level of the source voltage **1099** to the logic level and provide a zero output when push button **1098** is actuated. The set output of flip-flop **1090** enables NAND gate **1088** and drives the lamp interface **1094** to maintain the illumination of lamp **1096**. When the address bus **1083** outputs a binary word A which is equal to the push button address word B, the comparator equality (A=B) output goes high to drive the output of NAND gate **1088** low to provide a true car call CCLZ in the proper time slot. When the call is answered, a reset will appear on the reset bus **1089** when the comparator **1082** is addressed, driving the output of NAND gate **1086** low to reset the call memory **1090**, which removes the enable from NAND gate **1088** and removes the drive from the lamp interface.

A call set appearing on the call set bus in the time slot of circuit **1080**, which originates either in the penthouse, or in an auxiliary push button station, as hereinbefore described, will drive the output of NAND gate **1084** low when the comparator **1082** provides an

equality signal in the same time slot, which functions in the same manner as when the interface **1092** provides a low output in response to the push button **1098** being depressed.

In summary, there has been disclosed a new and improved elevator system having an elevator car which includes main and auxiliary car call stations. The labor require to interconnect the two car stations is substantially reduced, as only a few wires are required between the stations regardless of the number of landings in the associated structure. Instead of connecting the pushbuttons of the two stations in parallel, to obtain simultaneous indication on both stations as to which car calls have been registered, parallel interconnection is simulated by serializing the calls of the two stations and providing a car call set signal for each station which is responsive to the car calls registered on the other station. The car call set signal applied to the main and the auxiliary car stations may be a common signal representing the combined car calls from the two stations such that a call on the car call set line which was registered on one station will automatically set the associated car call memory element of the other station. The car call memory element of the station on which the call was initiated is not affected by the associated signal on the set line since its car call memory element is already set. Or, two separate car call set signals may be used, each of which include only the car calls set on one of the stations. Thus, the serialized car call signal from the main car station is used to set car calls on the auxiliary car station, and a serialized car signal from the auxiliary car station is used to set car calls on the main car station.

I claim as my invention:

1. An elevator system, comprising:

a structure having a plurality of landings,

an elevator car mounted for movement relative to said structure to serve the landings,

car control means including a main car station and an auxiliary car station disposed in said elevator car,

means for registering calls for service to said landings from each of said main and auxiliary car stations,

means storing the calls registered on said main and auxiliary car stations in said elevator car,

means serializing the calls for service registered on said main and auxiliary stations,

and serial call set means reyponsive to said serialized calls for service, providing a serial car call set signal for the main car station and a serial car call set signal for the auxiliary car station which set a call initiated on one of the car stations on the other of the car stations, to simulate parallel interconnection of the means for registering calls for service from the main and auxiliary car stations.

2. The elevator system of claim **1** wherein the means which serializes the calls for service includes first means for serializing the calls from the main car station, second means for serializing the calls from the auxiliary car station, and third means responsive to said first and second means providing a single serial signal which includes all calls for service registered on the main and auxiliary car stations, and wherein the serial call set means is responsive to the single serial signal provided by said third means to provide like car call set signals for the main and auxiliary car stations.

3. The elevator system of claim **2** including means for providing car call set signals from a point remote from the elevator car, wherein the third means is responsive to said remote car call set signals, incorporating the remote car call set signals into the single serial signal.

4. The elevator system of claim **1** including scan counter means which repetitively counts through a predetermined group of numbers to provide a plurality of timing signals which divide repetitive periods of time into a plurality of scan slots, with the landings of the structure being assigned to different scan slots, wherein the means serializing the calls for service places the calls into the scan slots assigned to the landings they are associated with.

5. The elevator system of claim **4** including means for providing car call reset signals in the scan slots associated with the landings, and means modifying the serial call set signals provided by the serial car call set means in response to said reset signals, to eliminate a car call set signal from a time slot in which a car call reset signal appears.

6. The elevator system of claim **1** including indicating means which visually indicates calls initiated on each of the main and auxiliary car stations, wherein the serial call set signal when setting a call on one of the stations also actuates its associated indicating means.

7. The elevator system of claim **1** wherein the means for registering calls for service from each of the main and auxiliary car stations includes pushbuttons.

8. The elevator system of claim **1** wherein the means which serializes the calls for service includes first means for serializing the calls from the main car station and second means for serializing the calls from the auxiliary car station, and wherein the serial call set means is responsive to the first and second means to provide a serial car call set signal for main car station responsive to the second means and a serial car call set signal for the auxiliary car station responsive to the first means.

* * * * *