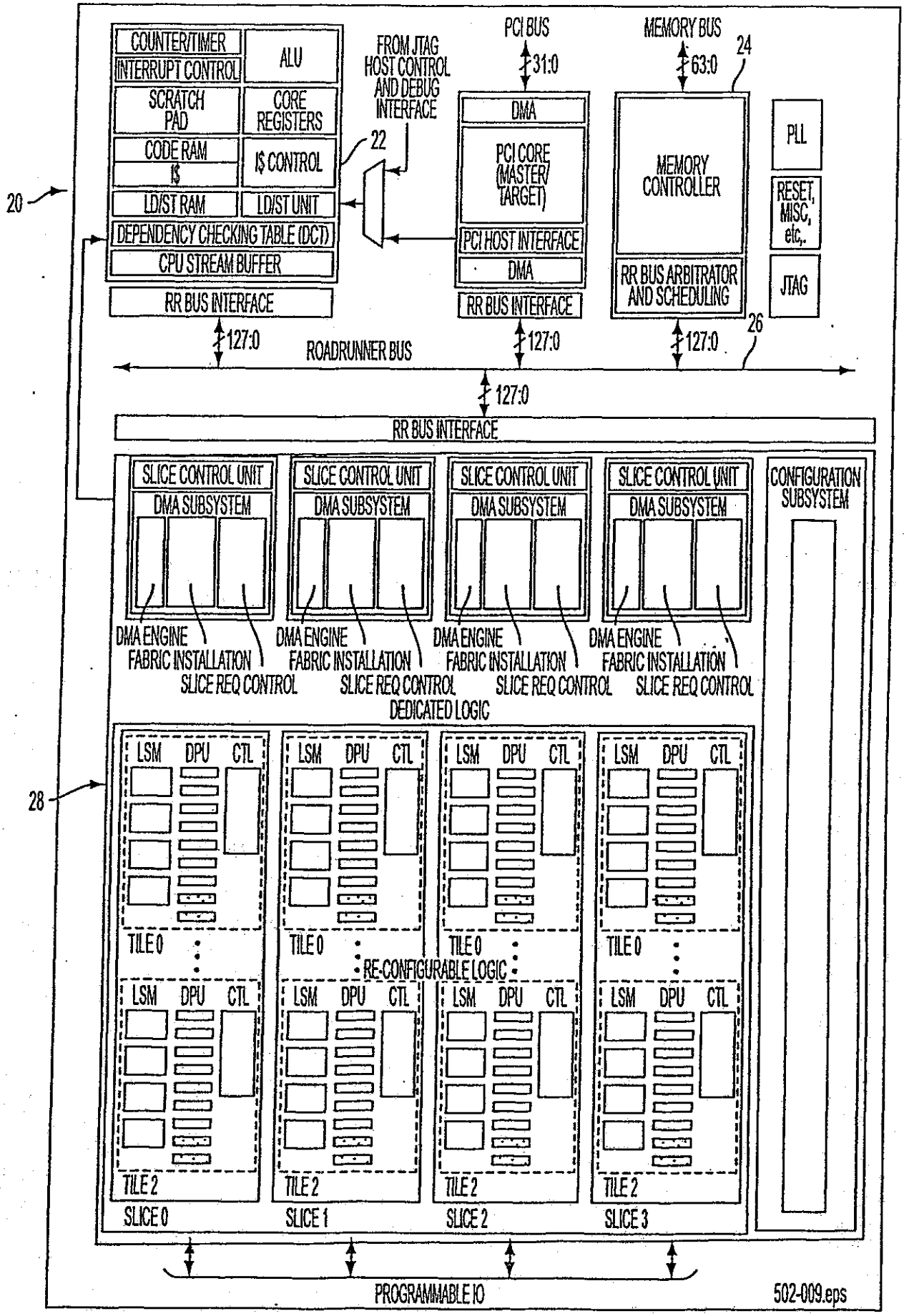

COUNTER/TIMER
INTERRUPT CONTROL
ALU
SCRATCH PAD
CORE REGISTERS
CODE RAM I$
I$ CONTROL
LD/ST RAM
LD/ST UNIT
DEPENDENCY CHECKING TABLE (DCT)
CPU STREAM BUFFER
RR BUS INTERFACE
FROM JTAG HOST CONTROL AND DEBUG INTERFACE
22
20
PCI BUS
31:0
DMA
PCI CORE (MASTER/ TARGET)
PCI HOST INTERFACE
DMA
RR BUS INTERFACE
MEMORY BUS
63:0
24
MEMORY CONTROLLER
RR BUS ARBITRATOR AND SCHEDULING
PLL
RESET, MISC, etc,.
JTAG
127:0
ROADRUNNER BUS
127:0
127:0
26
127:0
RR BUS INTERFACE
SLICE CONTROL UNIT
DMA SUBSYSTEM
CONFIGURATION SUBSYSTEM
DMA ENGINE
FABRIC INSTALLATION
SLICE REQ CONTROL
DEDICATED LOGIC
28
LSM
DPU
CTL
TILE 0
TILE 2
RE-CONFIGURABLE LOGIC
SLICE 0
SLICE 1
SLICE 2
SLICE 3
PROGRAMMABLE IO
502-009.eps

FIG. 1

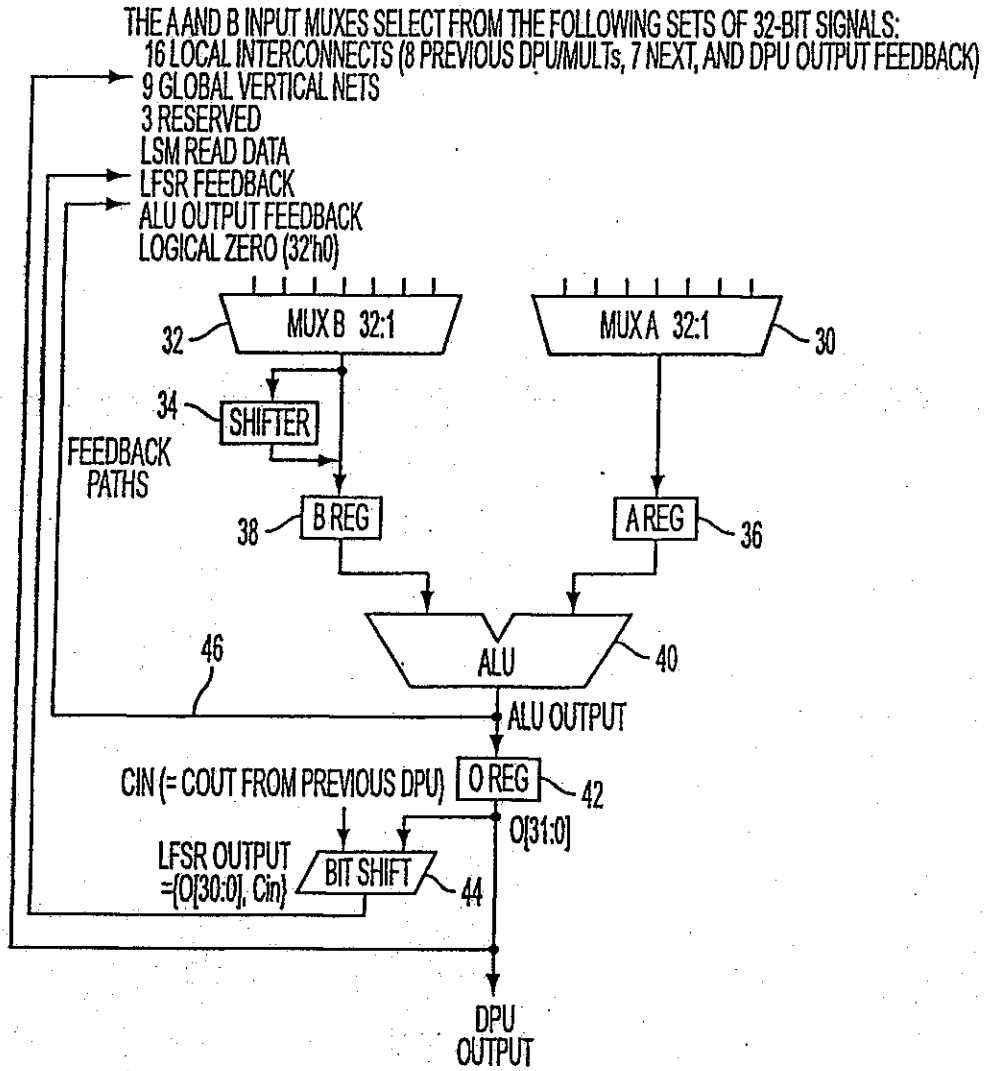

THE A AND B INPUT MUXES SELECT FROM THE FOLLOWING SETS OF 32-BIT SIGNALS:
16 LOCAL INTERCONNECTS (8 PREVIOUS DPU/MULTs, 7 NEXT, AND DPU OUTPUT FEEDBACK)
9 GLOBAL VERTICAL NETS
3 RESERVED
LSM READ DATA
LFSR FEEDBACK
ALU OUTPUT FEEDBACK
LOGICAL ZERO (32'h0)
MUX B 32:1
32
MUX A 32:1
30
34
SHIFTER
FEEDBACK PATHS
38
B REG
A REG
36
ALU
40
46
ALU OUTPUT
CIN (= COUT FROM PREVIOUS DPU)
O REG
42
O[31:0]
LFSR OUTPUT ={O[30:0], Cin}
BIT SHIFT
44
DPU OUTPUT

FIG. 2

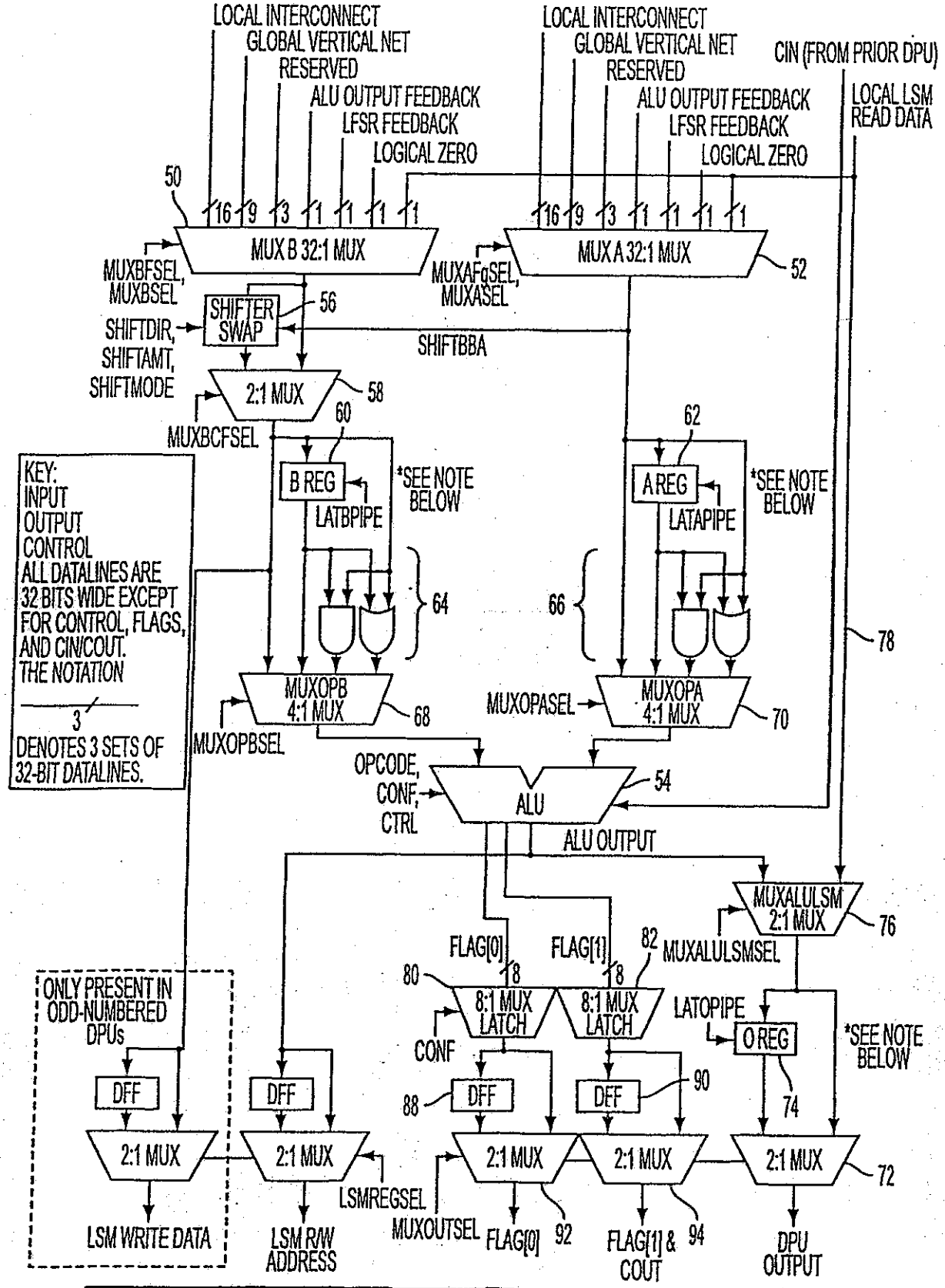

LOCAL INTERCONNECT
GLOBAL VERTICAL NET
RESERVED
ALU OUTPUT FEEDBACK
LFSR FEEDBACK
LOGICAL ZERO
CIN (FROM PRIOR DPU)
LOCAL LSM READ DATA
50
16 9 3 1 1 1 1
MUX B 32:1 MUX
MUX A 32:1 MUX
52
MUXBFSEL, MUXBSEL
MUXAFSEL, MUXASEL
SHIFTER SWAP
56
SHIFTDIR, SHIFTAMT, SHIFTMODE
SHIFTBBA
2:1 MUX
58
MUXBCFSEL
60
62
B REG
A REG
*SEE NOTE BELOW
LATBPIPE
LATAPIPE
64
66
78
MUXOPB 4:1 MUX
MUXOPA 4:1 MUX
68
70
MUXOPBSEL
MUXOPASEL
OPCODE, CONF, CTRL
ALU
54
ALU OUTPUT
MUXALULSM 2:1 MUX
76
FLAG[0]
FLAG[1]
82
MUXALULSMSEL
80
8:1 MUX LATCH
LATOPIPE
O REG
CONF
DFF
88
90
74
ONLY PRESENT IN ODD-NUMBERED DPUs
LSMREGSEL
MUXOUTSEL
92
94
72
LSM WRITE DATA
LSM R/W ADDRESS
FLAG[0]
FLAG[1] & COUT
DPU OUTPUT
KEY:
INPUT
OUTPUT
CONTROL
ALL DATALINES ARE 32 BITS WIDE EXCEPT FOR CONTROL, FLAGS, AND CIN/COUT.
THE NOTATION
3
DENOTES 3 SETS OF 32-BIT DATALINES.
* THE 'A', 'B' AND 'O' REGISTERS CAN ALSO BE LOADED FROM CONFIGURATION AND THE ARC PROCESSOR. ALL FLIP-FLOPS ARE INITIALIZED TO ZERO UPON ACTIVATION, EXCEPT THE 'A', 'B' AND 'O' REGISTERS, WHICH ARE INITIALIZED WITH CONFIGURATION DATA.
502-171.eps

FIG. 3

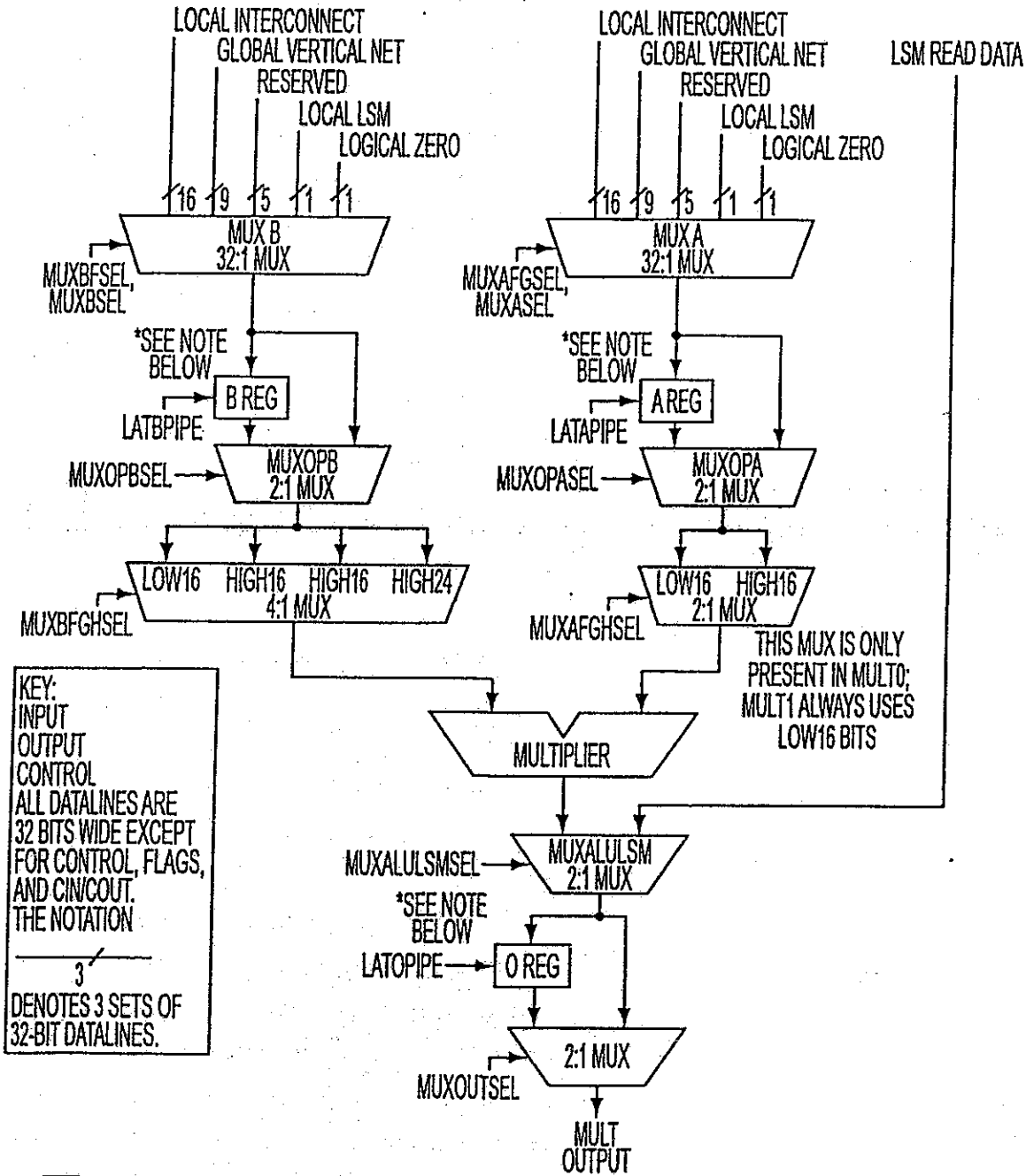

LOCAL INTERCONNECT
GLOBAL VERTICAL NET
RESERVED
LOCAL LSM
LOGICAL ZERO
16
9
5
1
1
MUX B
32:1 MUX
MUXBFSEL,
MUXBSEL
*SEE NOTE BELOW
B REG
LATBPIPE
MUXOPBSEL
MUXOPB
2:1 MUX
LOW16
HIGH16
HIGH16
HIGH24
4:1 MUX
MUXBFGHSEL
LOCAL INTERCONNECT
GLOBAL VERTICAL NET
RESERVED
LOCAL LSM
LOGICAL ZERO
LSM READ DATA
MUX A
32:1 MUX
MUXAFGSEL,
MUXASEL
*SEE NOTE BELOW
A REG
LATAPIPE
MUXOPASEL
MUXOPA
2:1 MUX
LOW16
HIGH16
2:1 MUX
MUXAFGHSEL
THIS MUX IS ONLY PRESENT IN MULT0; MULT1 ALWAYS USES LOW16 BITS
KEY:
INPUT
OUTPUT
CONTROL
ALL DATALINES ARE 32 BITS WIDE EXCEPT FOR CONTROL, FLAGS, AND CIN/COUT.
THE NOTATION
3
DENOTES 3 SETS OF 32-BIT DATALINES.
MULTIPLIER
MUXALULSMSEL
MUXALULSM
2:1 MUX
*SEE NOTE BELOW
LATOPIPE
O REG
2:1 MUX
MUXOUTSEL
MULT OUTPUT
* THE 'A', 'B' AND 'O' REGISTERS CAN ALSO BE LOADED FROM CONFIGURATION AND THE ARC PROCESSOR. THE 'A', 'B' AND 'O' REGISTERS ARE INITIALIZED WITH CONFIGURATION DATA.

FIG. 4

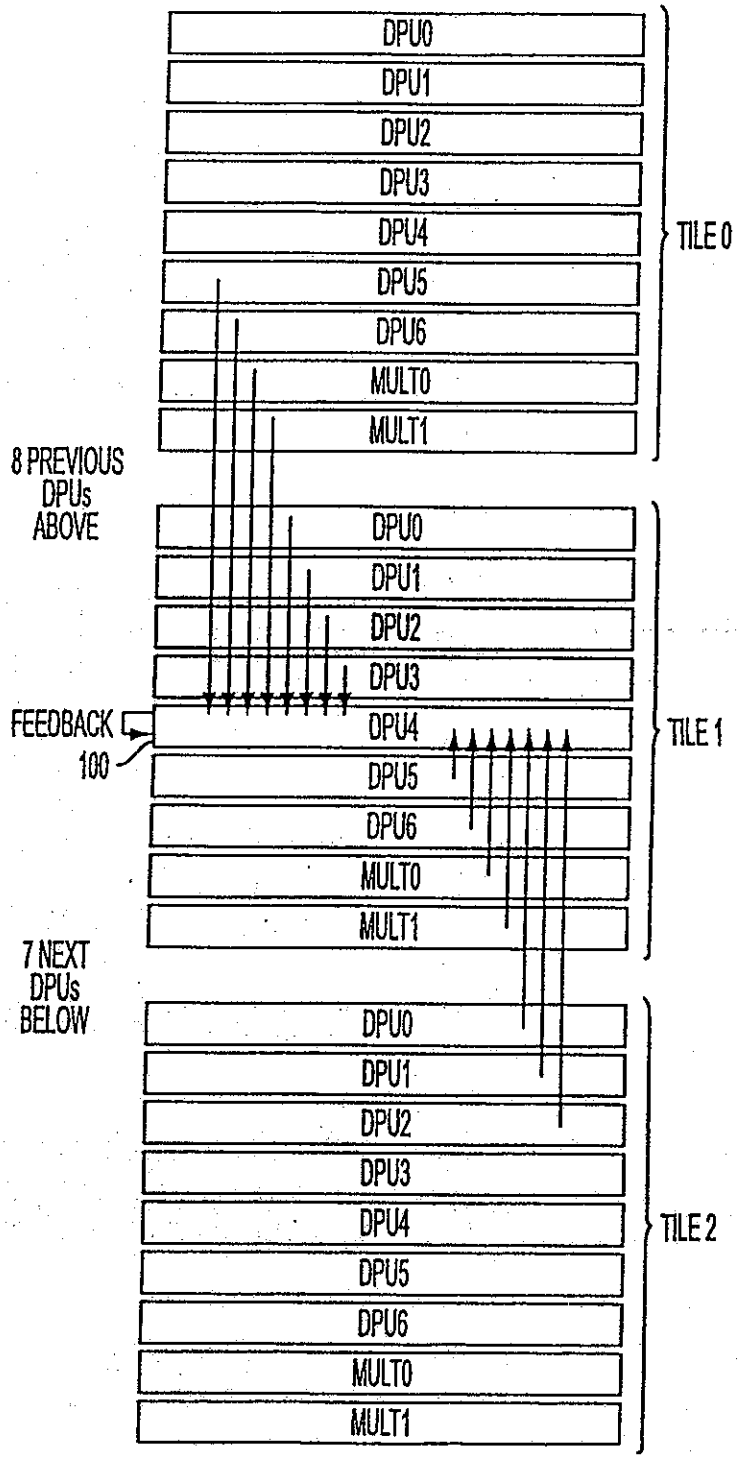

DPU0
DPU1
DPU2
DPU3
DPU4
DPU5
DPU6
MULT0
MULT1
TILE 0
8 PREVIOUS DPUs ABOVE
DPU0
DPU1
DPU2
DPU3
FEEDBACK
100
DPU4
DPU5
DPU6
MULT0
MULT1
TILE 1
7 NEXT DPUs BELOW
DPU0
DPU1
DPU2
DPU3
DPU4
DPU5
DPU6
MULT0
MULT1
TILE 2

FIG. 5

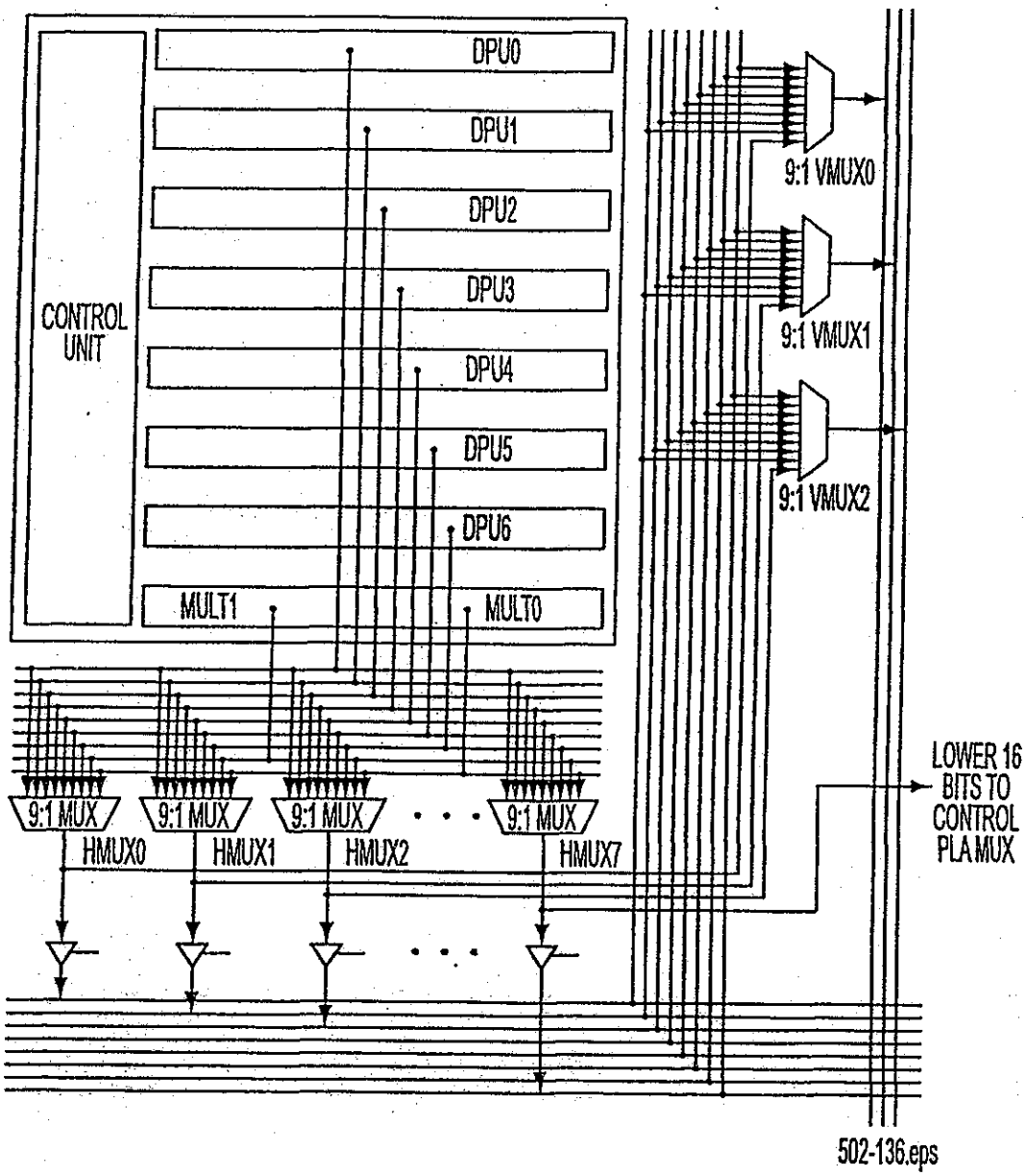

CONTROL UNIT
DPU0
DPU1
DPU2
DPU3
DPU4
DPU5
DPU6
MULT1
MULT0
9:1 VMUX0
9:1 VMUX1
9:1 VMUX2
9:1 MUX
9:1 MUX
9:1 MUX
9:1 MUX
HMUX0
HMUX1
HMUX2
HMUX7
LOWER 16 BITS TO CONTROL PLA MUX
502-136.eps

FIG. 6

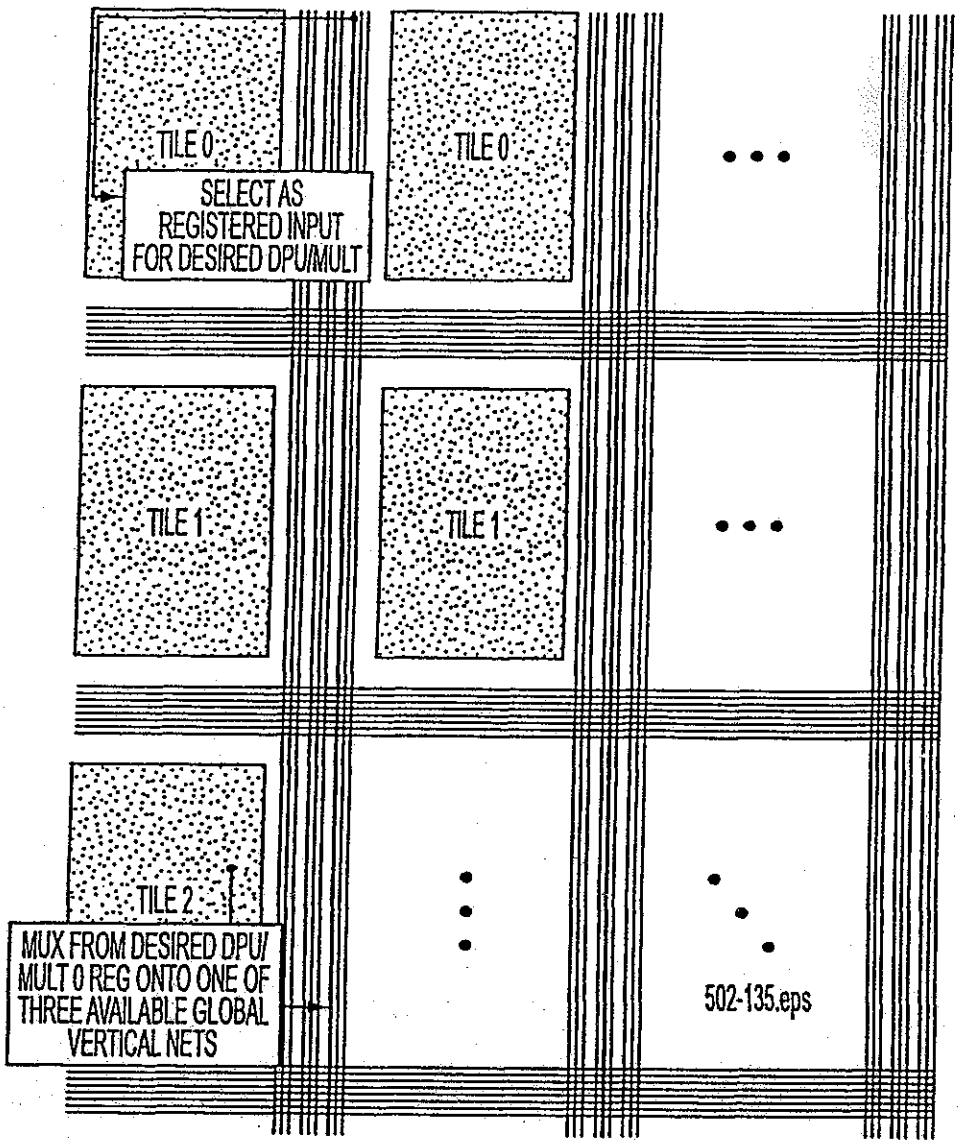

TILE 0
SELECT AS REGISTERED INPUT FOR DESIRED DPU/MULT
TILE 0
TILE 1
TILE 1
TILE 2
MUX FROM DESIRED DPU/ MULT 0 REG ONTO ONE OF THREE AVAILABLE GLOBAL VERTICAL NETS
502-135.eps

FIG. 7

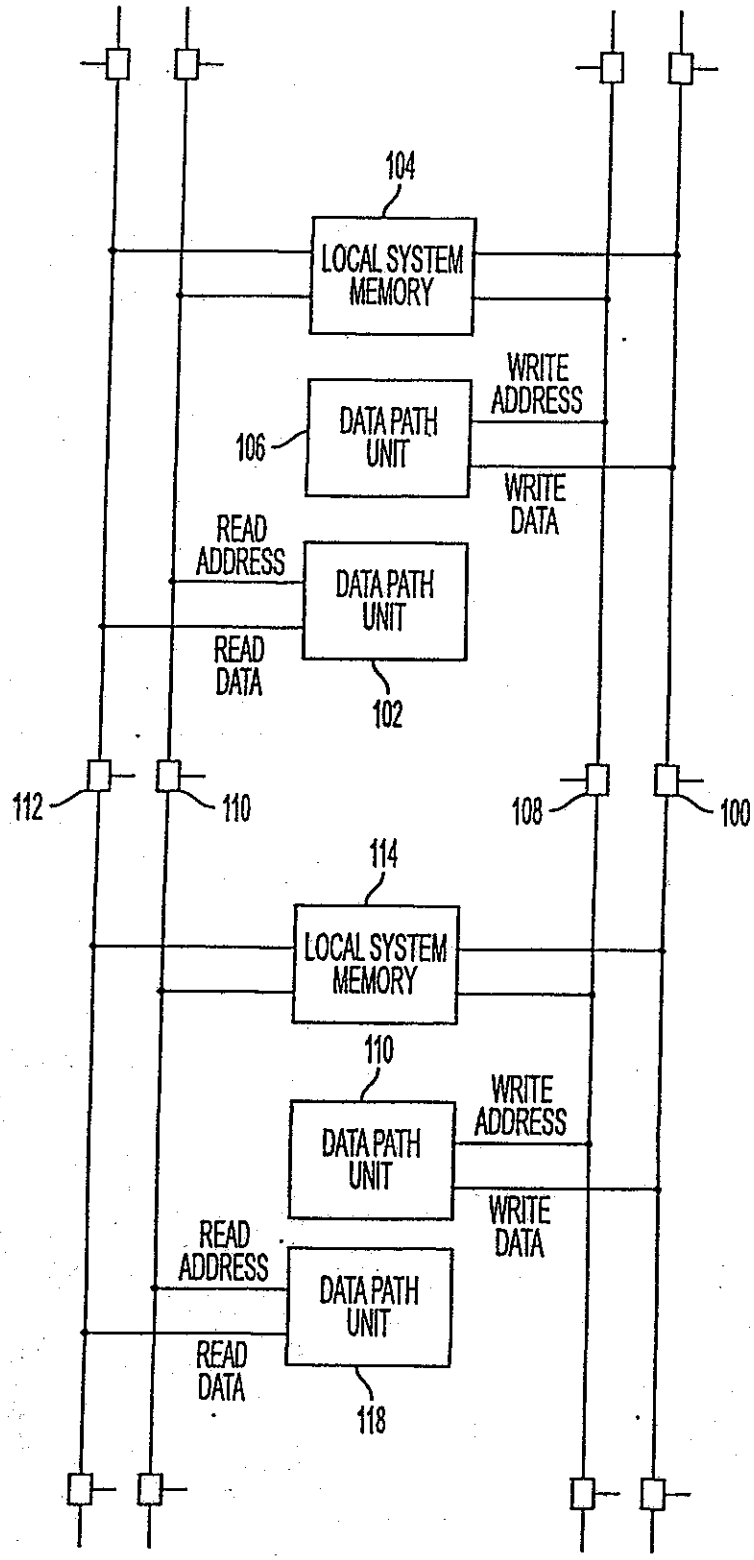

104
LOCAL SYSTEM MEMORY
WRITE ADDRESS
106
DATA PATH UNIT
WRITE DATA
READ ADDRESS
DATA PATH UNIT
READ DATA
102
112
110
108
100
114
LOCAL SYSTEM MEMORY
110
WRITE ADDRESS
DATA PATH UNIT
WRITE DATA
READ ADDRESS
DATA PATH UNIT
READ DATA
118

FIG. 8

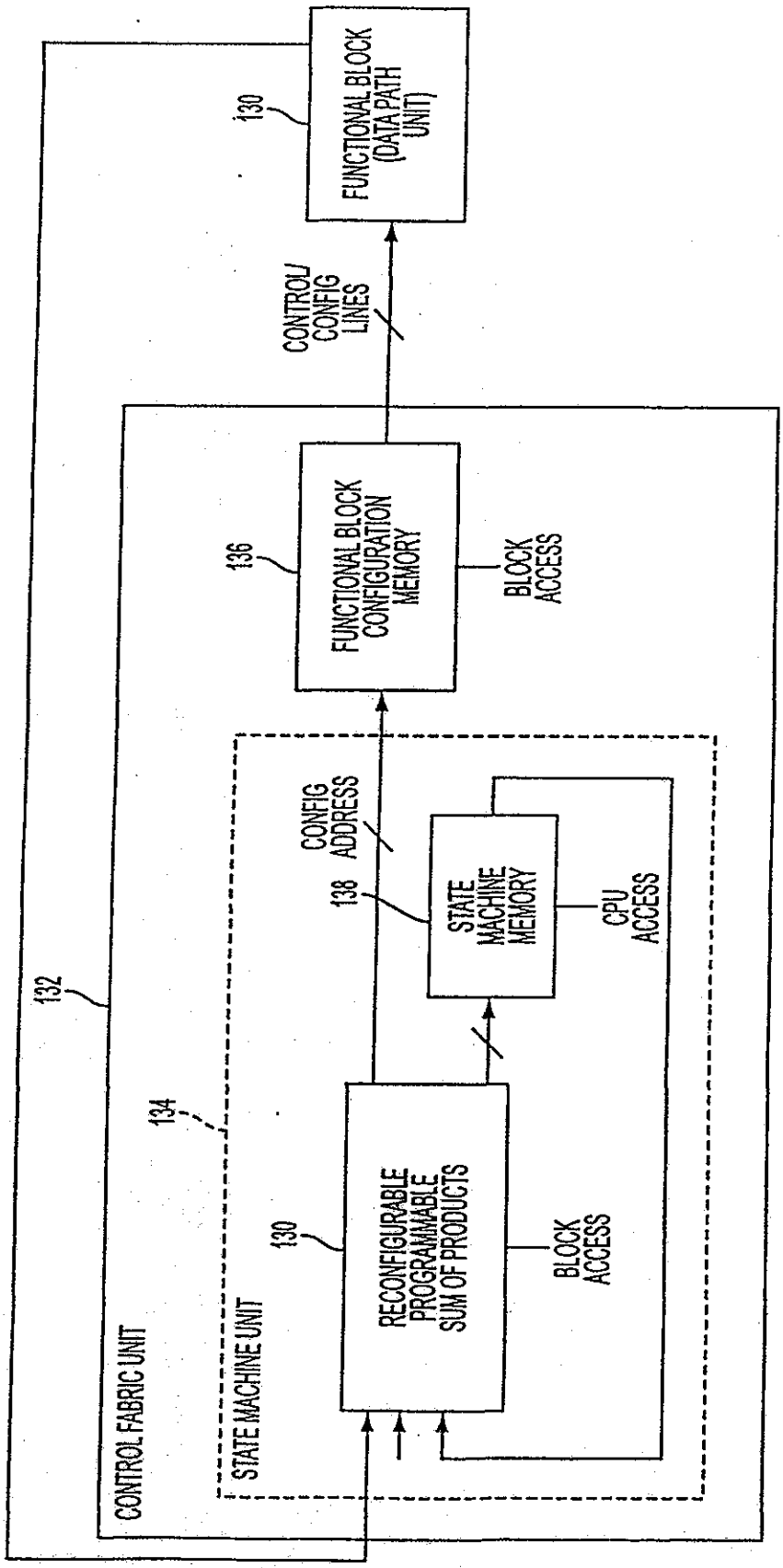

FUNCTIONAL BLOCK (DATA PATH UNIT)
130
CONTROL/ CONFIG LINES
FUNCTIONAL BLOCK CONFIGURATION MEMORY
136
BLOCK ACCESS
CONFIG ADDRESS
138
STATE MACHINE MEMORY
CPU ACCESS
132
134
CONTROL FABRIC UNIT
STATE MACHINE UNIT
130
RECONFIGURABLE PROGRAMMABLE SUM OF PRODUCTS
BLOCK ACCESS

FIG. 9

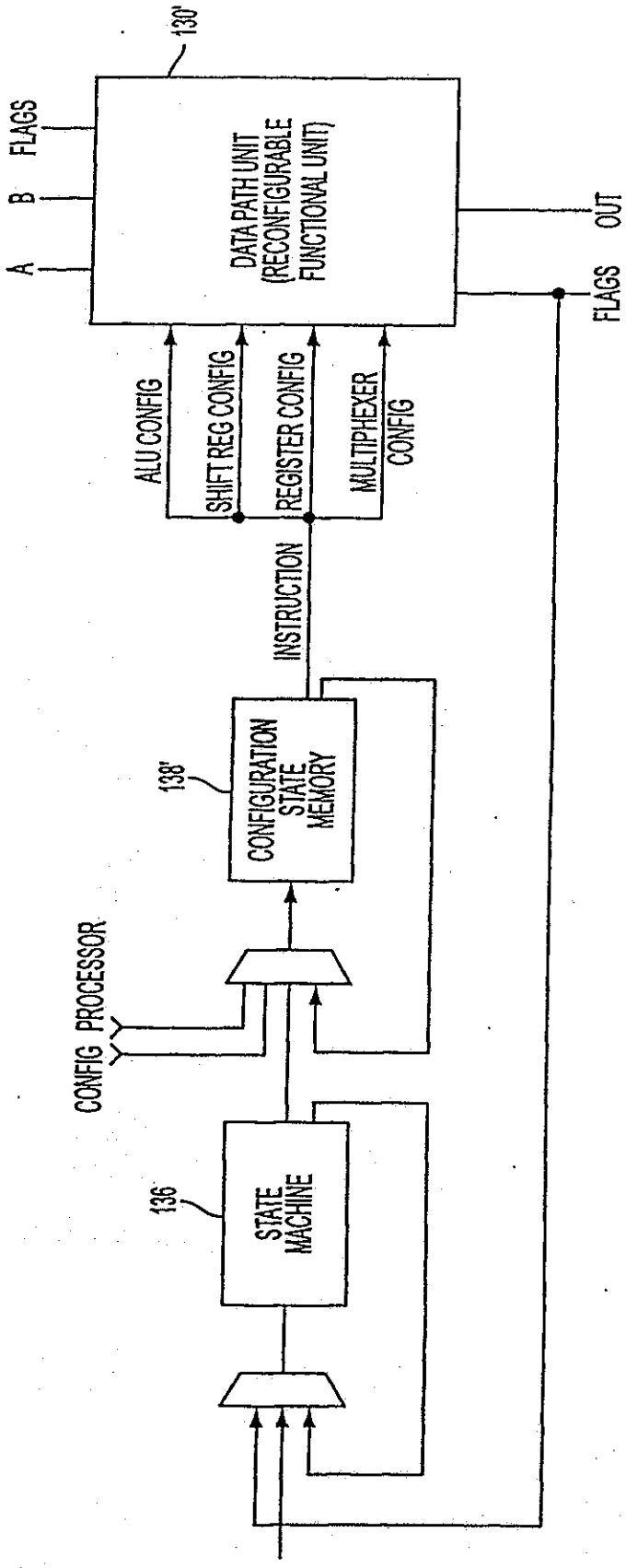

130'
DATA PATH UNIT (RECONFIGURABLE FUNCTIONAL UNIT)
A
B
FLAGS
OUT
FLAGS
ALU CONFIG
SHIFT REG CONFIG
REGISTER CONFIG
MULTIPHEXER CONFIG
INSTRUCTION
138'
CONFIGURATION STATE MEMORY
CONFIG PROCESSOR
136
STATE MACHINE

FIG. 10A

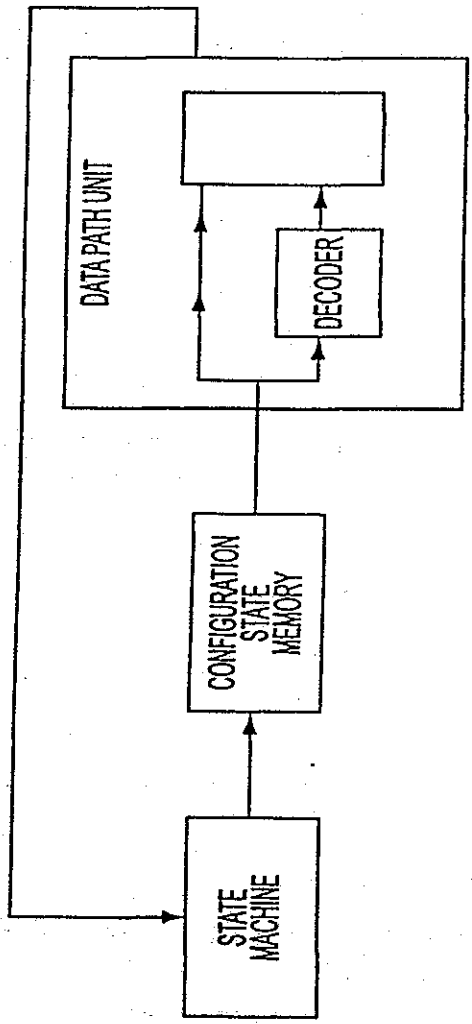

DATA PATH UNIT
DECODER
CONFIGURATION STATE MEMORY
STATE MACHINE

FIG. 10B

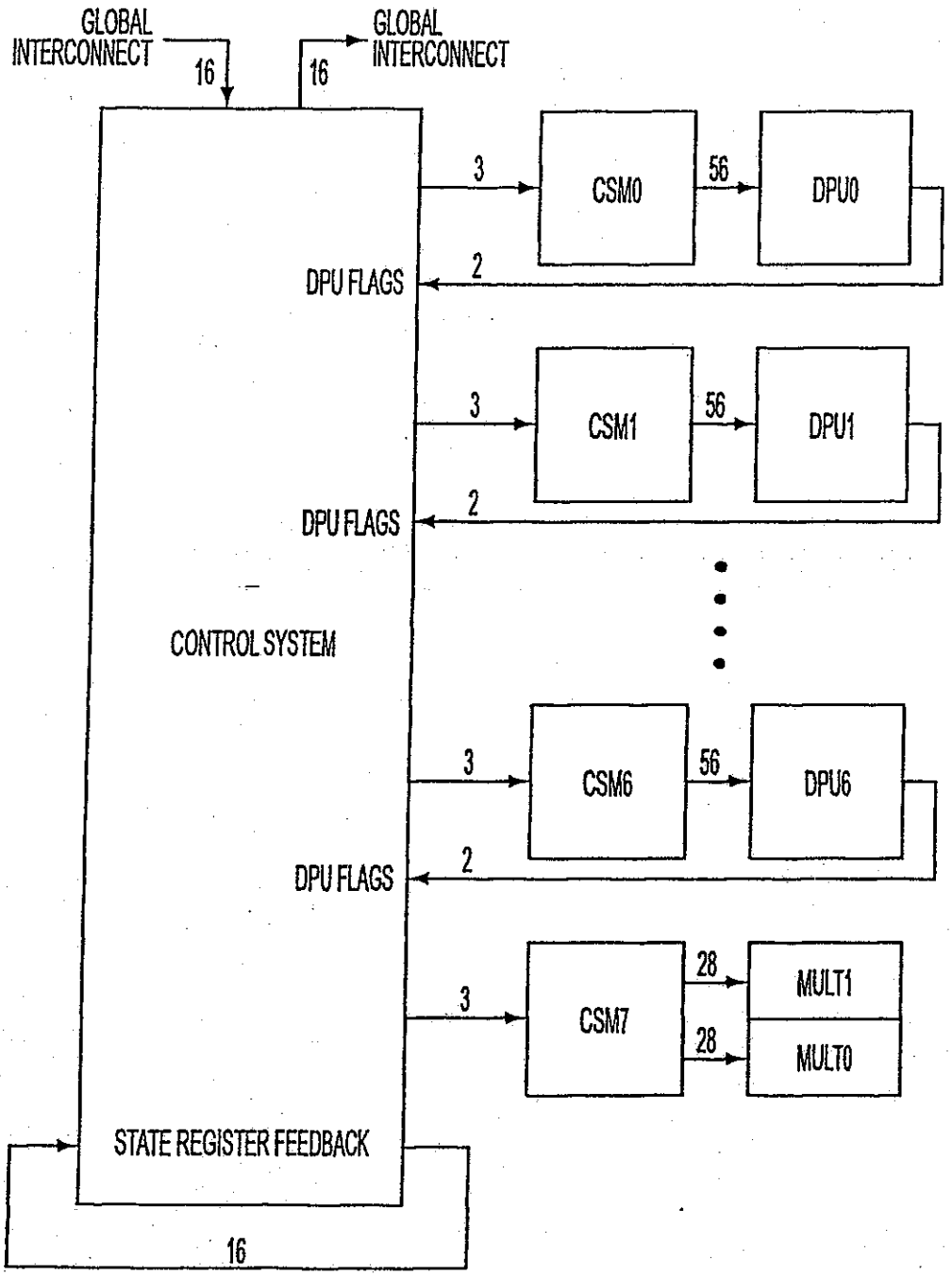

GLOBAL INTERCONNECT
16
16
GLOBAL INTERCONNECT
3
CSM0
56
DPU0
DPU FLAGS
2
3
CSM1
56
DPU1
DPU FLAGS
2
CONTROL SYSTEM
3
CSM6
56
DPU6
DPU FLAGS
2
3
CSM7
28
MULT1
28
MULT0
STATE REGISTER FEEDBACK
16

FIG. 11

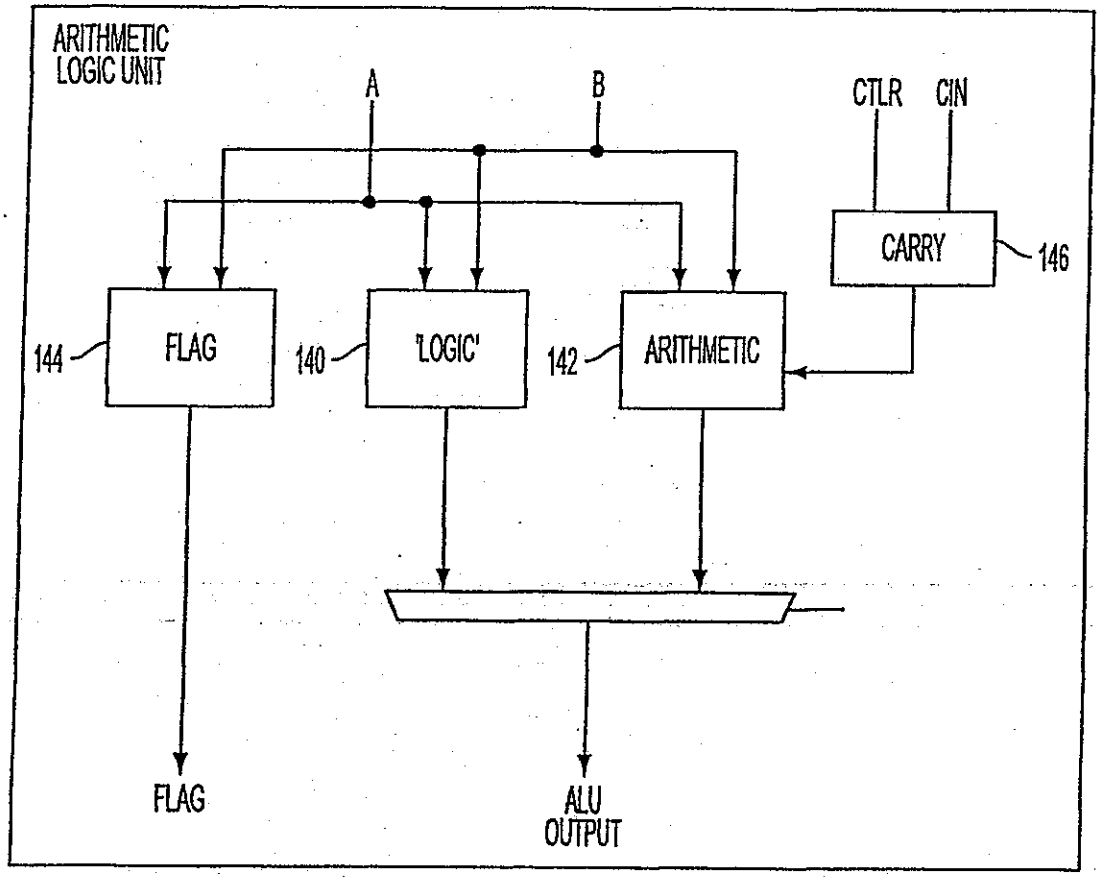

ARITHMETIC LOGIC UNIT
A
B
CTLR
CIN
CARRY
146
144
FLAG
140
'LOGIC'
142
ARITHMETIC
FLAG
ALU OUTPUT

FIG. 12

OPCODE OF DPU AT A GLANCE		
OPCODE	ENCODING(HEX)	DESCRIPTION
ADD	00	ADDITION
ADD16	80	TWO PARALLEL 16 BIT ADDITION
ADDC	10	ADDITION WITH CARRY IN
ADDCNT	18	ADDITION WITH CONTROL CARRY IN
SUB	09	SUBTRACTION
SUB16	89	TWO PARALLEL 16 BIT SUBTRACTION
SUBC	11	SUBTRACTION WITH CARRY IN
SUBCNT	19	SUBTRACTION WITH CONTROL CARRY IN
SADD	20	SATURATION VERSION OF ADD
SADD16	a0	TWO PARALLEL 16 BIT SAT. ADDITION
SADDCNT	38	SATURATION VERSION OF ADDCNT
SSUB	29	SATURATION VERSION OF SUB
SSUB16	a9	TWO PARALLEL 16 BIT SAT. SUBTRACTION
SSUBCNT	39	SATURATION VERSION OF SUBCNT
INC	0c	INCREMENT B
DEC	06	DECREMENT
NEG	0d	NEGATE B
ABS	2d	ABSOLUTE B
ABS16	ad	TWO PARALLEL 16 BIT ABSOLUTE
CSUB	49	CONDITIONAL SUBTRACTION
AND	60	BITWISE AND
OR	62	BITWISE OR
NAND	61	BITWISE NAND
NOR	63	BITWISE NOR
XOR	64	BITWISE XOR
XNOR	65	BITWISE XNOR
PASSA	66	PASS A
PASSB	6e	PASS B
NOTA	67	INVERT A
NOTB	6f	INVERT B
MIN	0b	MINIMUM
MIN16	8b	TWO PARALLEL 16 BIT MINIMUM
MAX	2b	MAXIMUM
MAX16	ab	TWO PARALLEL 16 BIT MAXIMUM
PENC	7f	PRIORITY ENCODE
MUXBBA	ce	MULTIPLEX B BY A
SHIFTBBA	de	SHIFT B BY A

FIG. 13A

ESSENTIAL BIS FOR FAST DECODING								
OPCODE				Cmux1	Cmux0	Amux1	Amux0	Bmux
ADD	0	0	0	0	0	0	0	0
ADD16	1	0	0	0	0	0	0	0
ADDC	0	0	0	1	0	0	0	0
ADDCNT	0	0	0	1	1	0	0	0
SUB	0	0	0	0	1	0	0	1
SUB16	1	0	0	0	1	0	0	1
SUBC	0	0	0	1	0	0	0	1
SUBCNT	0	0	0	1	1	0	0	1
SADD	0	0	1	0	0	0	0	0
SADD16	1	0	1	0	0	0	0	0
SADDCNT	0	0	1	1	1	0	0	0
SSUB	0	0	1	0	1	0	0	1
SSUB16	1	0	1	0	1	0	0	1
SSUBCNT	0	0	1	1	1	0	0	1
INC	0	0	0	0	1	1	0	0
DEC	0	0	0	0	0	1	1	0
NEG	0	0	0	0	1	1	0	1
ABS	0	0	1	0	1	1	0	1
ABS16	1	0	1	0	1	1	0	1
CSUB	0	1	0	0	1	0	0	1
AND	0	1	1	0	0	0	0	0
OR	0	1	1	0	0	0	1	0
NAND	0	1	1	0	0	0	0	1
NOR	0	1	1	0	0	0	1	1
XOR	0	1	1	0	0	1	0	0
XNOR	0	1	1	0	0	1	0	1
PASSA	0	1	1	0	0	1	1	0
PASSB	0	1	1	0	1	1	1	0
NOTA	0	1	1	0	0	1	1	1
NOTB	0	1	1	0	1	1	1	1
MIN	0	0	0	0	1	0	1	1
MIN16	1	0	0	0	1	0	1	1
MAX	0	0	1	0	1	0	1	1
MAX16	1	0	1	0	1	0	1	1
PENC	0	1	1	1	1	1	1	1
MUXBBA	1	1	0	0	1	1	1	0
SHIFTBBA	1	1	0	1	1	1	1	0

bmux: 0B 1B_
amux: 0A 1 20 3-1
cmux: 00 11 2cin 3cnt

OR 0positive 1negate
OR 0and 1 or 2xor 3pass

FIG. 13B

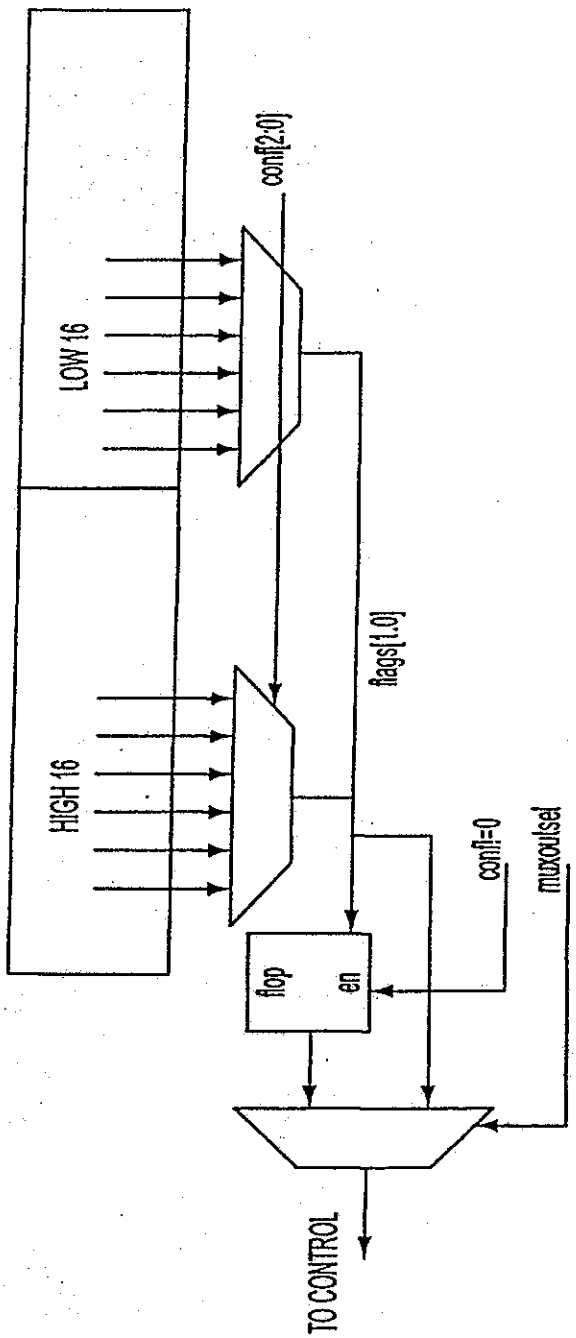

conf[2:0]
LOW 16
HIGH 16
flags[1:0]
flop
en
conf!=0
muxoutsel
TO CONTROL

FIG. 14

shiftdir*	shiftmode	OPERATION
0	000	32 BIT LOGICAL RIGHT SHIFT
1	000	32 BIT LOGICAL LEFT SHIFT
0	001	32 BIT ARITHMATIC RIGHT SHIFT
1	001	32 BIT ARITHMATIC LEFT SHIFT
X	010	(a) LOWER 16 BIT SIGN EXTEND TO 32 BIT (a)
X	011	(b) CONSTANT GENERATION FROM {shiftdir[0], shiftamt[4:0]}
X	100	(c) DUPLICATE LOWER 16 BIT TO UPPER 16 BIT
X	101	(d) DUPLICATE UPPER 16 BIT TO LOWER 16 BIT
X	110	(e) SWAP LOWER AND UPPER 16 BIT
0	111	(f) 16 BIT ARITHMATIC RIGHT SHIFT (IF THE shiftamt[4] IS ONE, THEN THE OUTPUT WILL BE UNPREDICTABLE)
1	111	(g) BYTE SWAP (THE 32 BIT CHANGES FROM {b0, b1, b2, b3} TO {b3,b2,b1, b0}

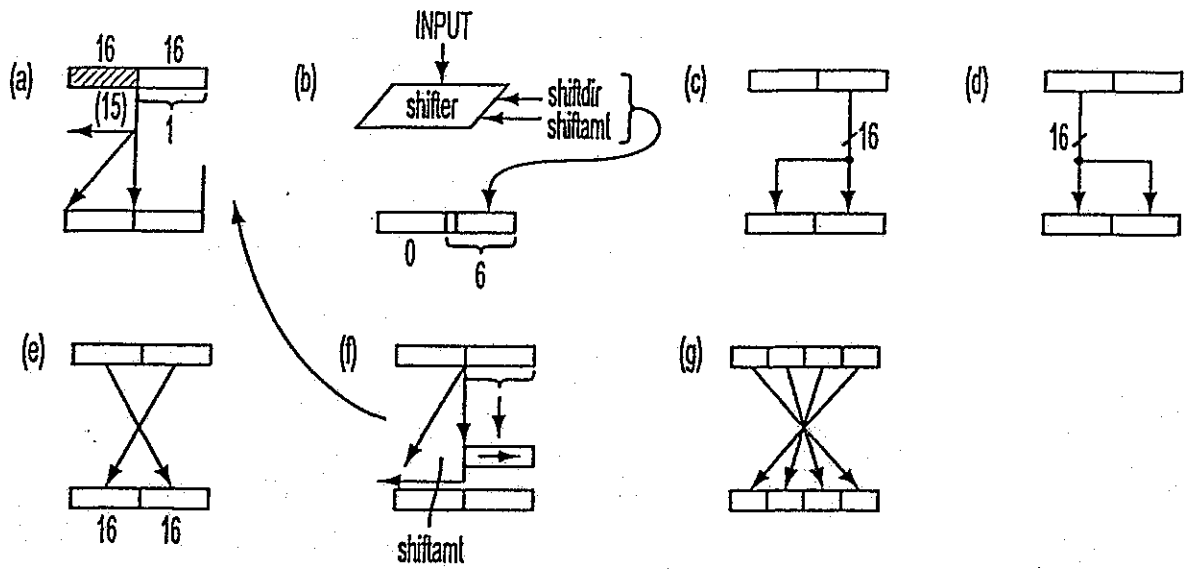


FIG. 15

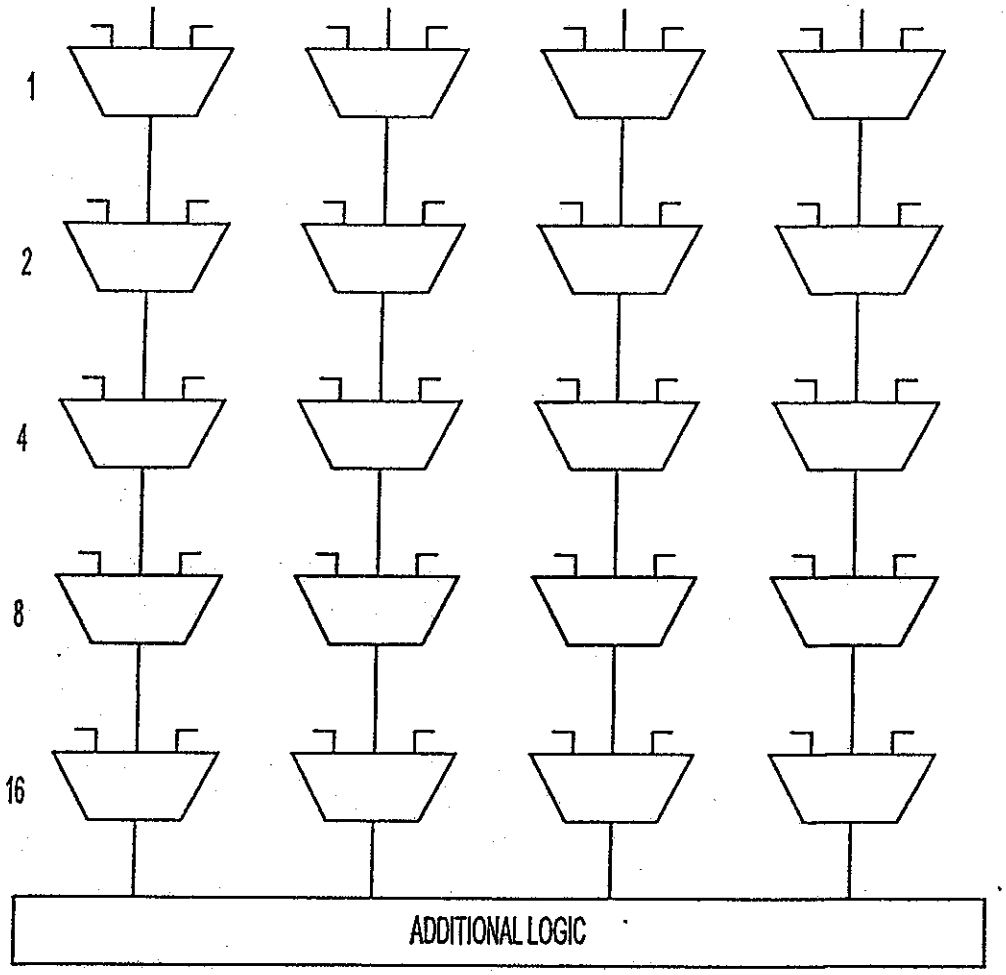

1
2
4
8
16
ADDITIONAL LOGIC

FIG. 16

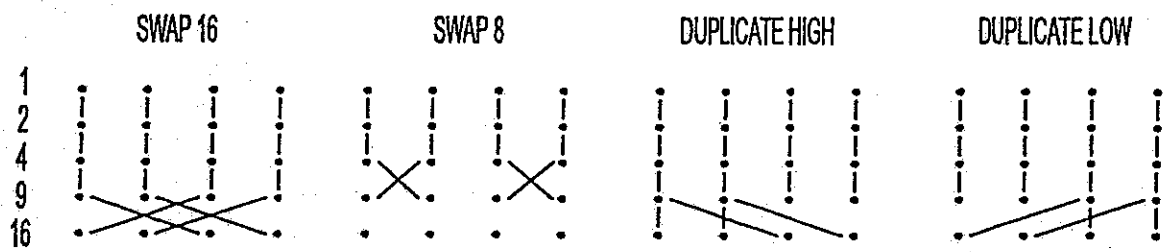

SWAP 16
SWAP 8
DUPLICATE HIGH
DUPLICATE LOW
1
2
4
9
16

FIG. 17

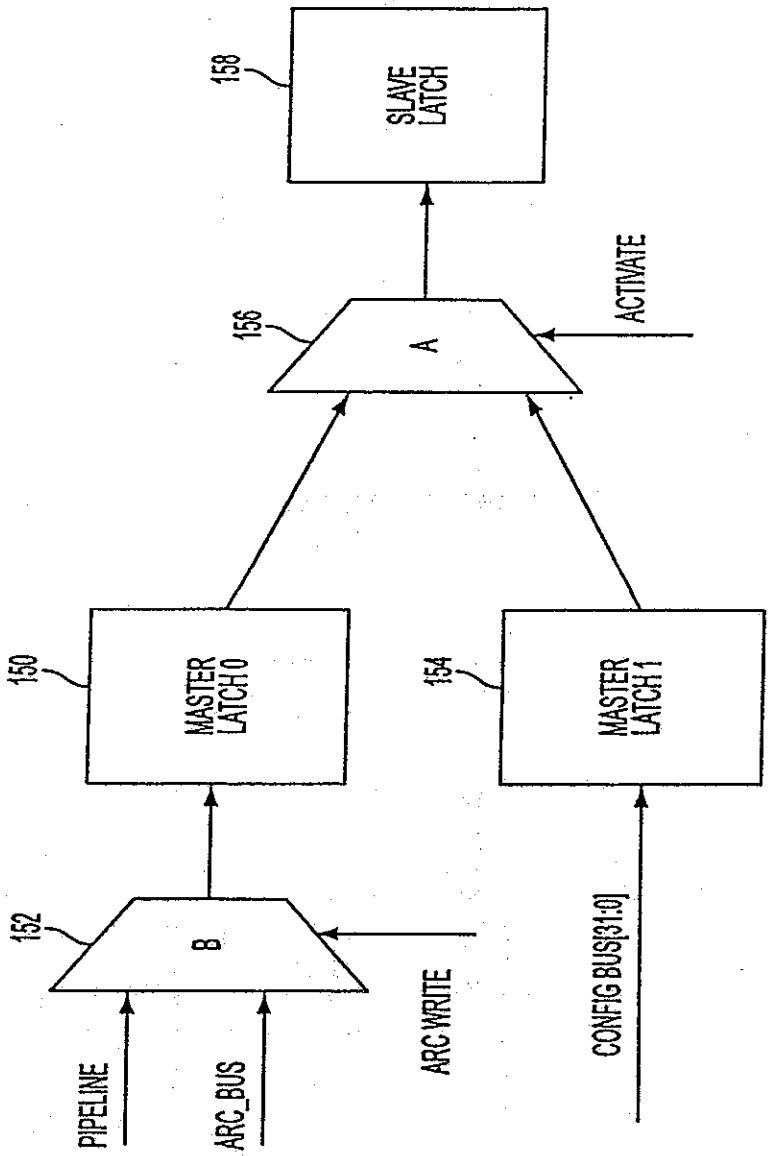

158
SLAVE LATCH
156
A
ACTIVATE
150
MASTER LATCH 0
154
MASTER LATCH 1
152
B
PIPELINE
ARC_BUS
ARC WRITE
CONFIG BUS[31:0]

FIG. 18

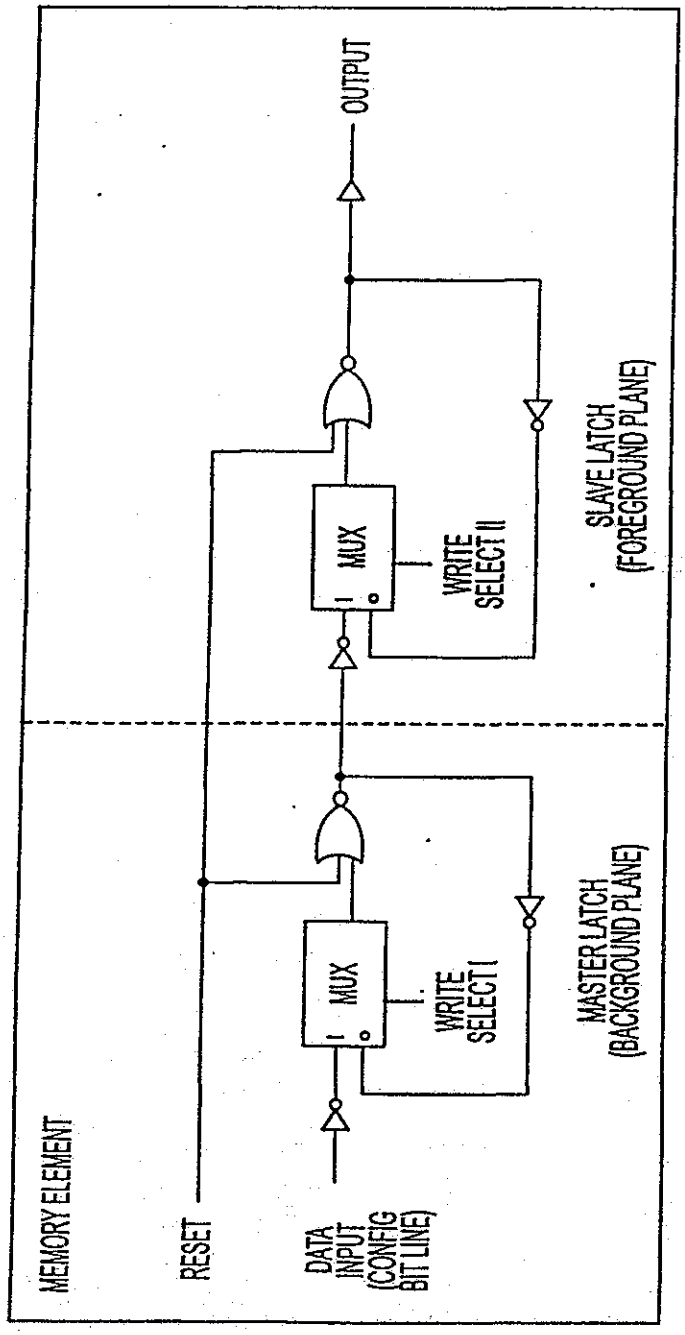

OUTPUT
MUX
WRITE SELECT II
SLAVE LATCH (FOREGROUND PLANE)
MUX
WRITE SELECT I
MASTER LATCH (BACKGROUND PLANE)
MEMORY ELEMENT
RESET
DATA INPUT (CONFIG BIT LINE)

FIG. 19

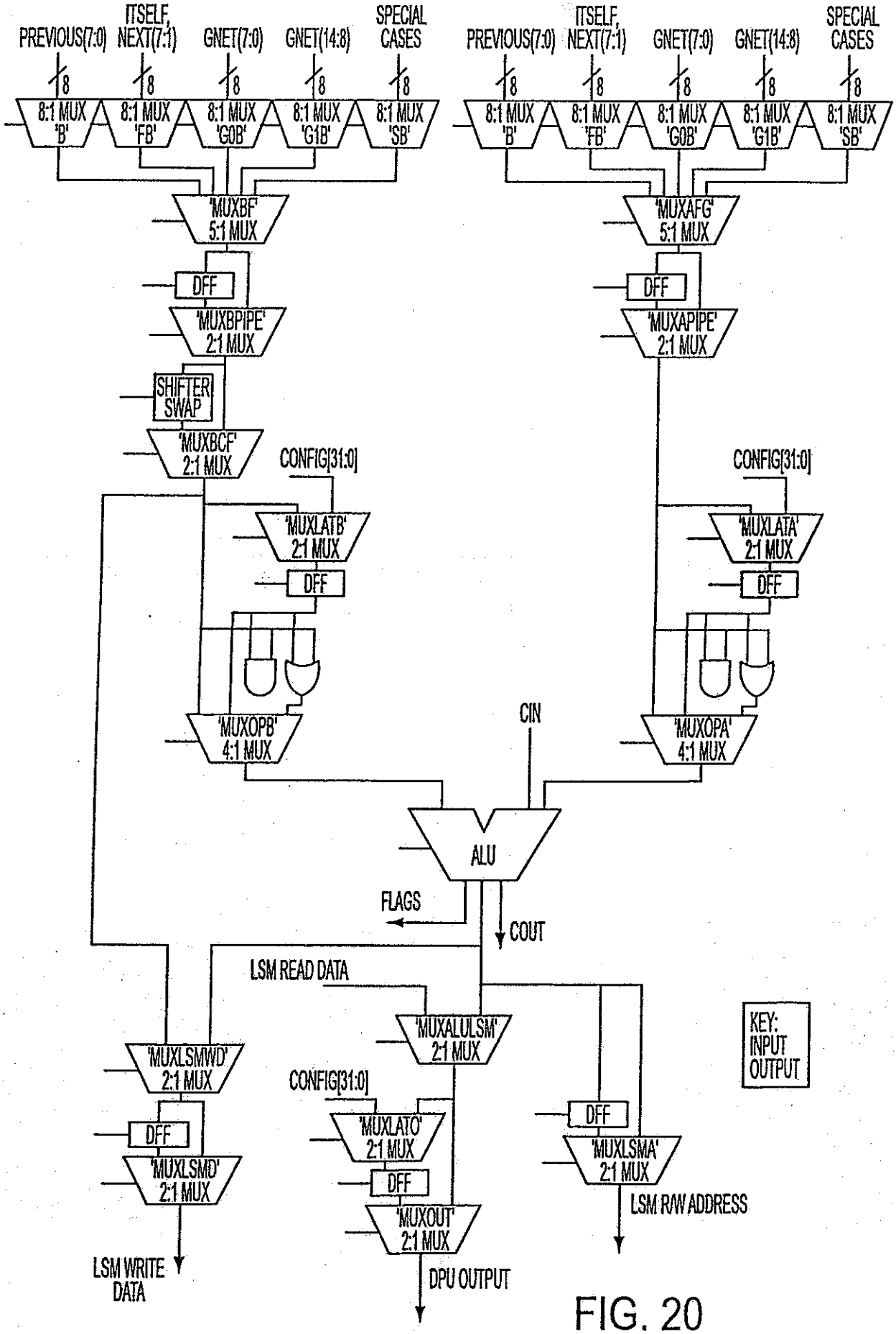

PREVIOUS(7:0)
ITSELF, NEXT(7:1)
GNET(7:0)
GNET(14:8)
SPECIAL CASES
8
8:1 MUX 'B'
8:1 MUX 'FB'
8:1 MUX 'G0B'
8:1 MUX 'G1B'
8:1 MUX 'SB'
'MUXBF' 5:1 MUX
'MUXAFG' 5:1 MUX
DFF
'MUXBPIPE' 2:1 MUX
'MUXAPIPE' 2:1 MUX
SHIFTER SWAP
'MUXBCF' 2:1 MUX
CONFIG[31:0]
'MUXLATB' 2:1 MUX
'MUXLATA' 2:1 MUX
'MUXOPB' 4:1 MUX
'MUXOPA' 4:1 MUX
CIN
ALU
FLAGS
COUT
LSM READ DATA
'MUXALULSM' 2:1 MUX
'MUXLSMWD' 2:1 MUX
'MUXLATO' 2:1 MUX
'MUXLSMD' 2:1 MUX
'MUXLSMA' 2:1 MUX
'MUXOUT' 2:1 MUX
LSM WRITE DATA
DPU OUTPUT
LSM R/W ADDRESS
KEY: INPUT OUTPUT

FIG. 20

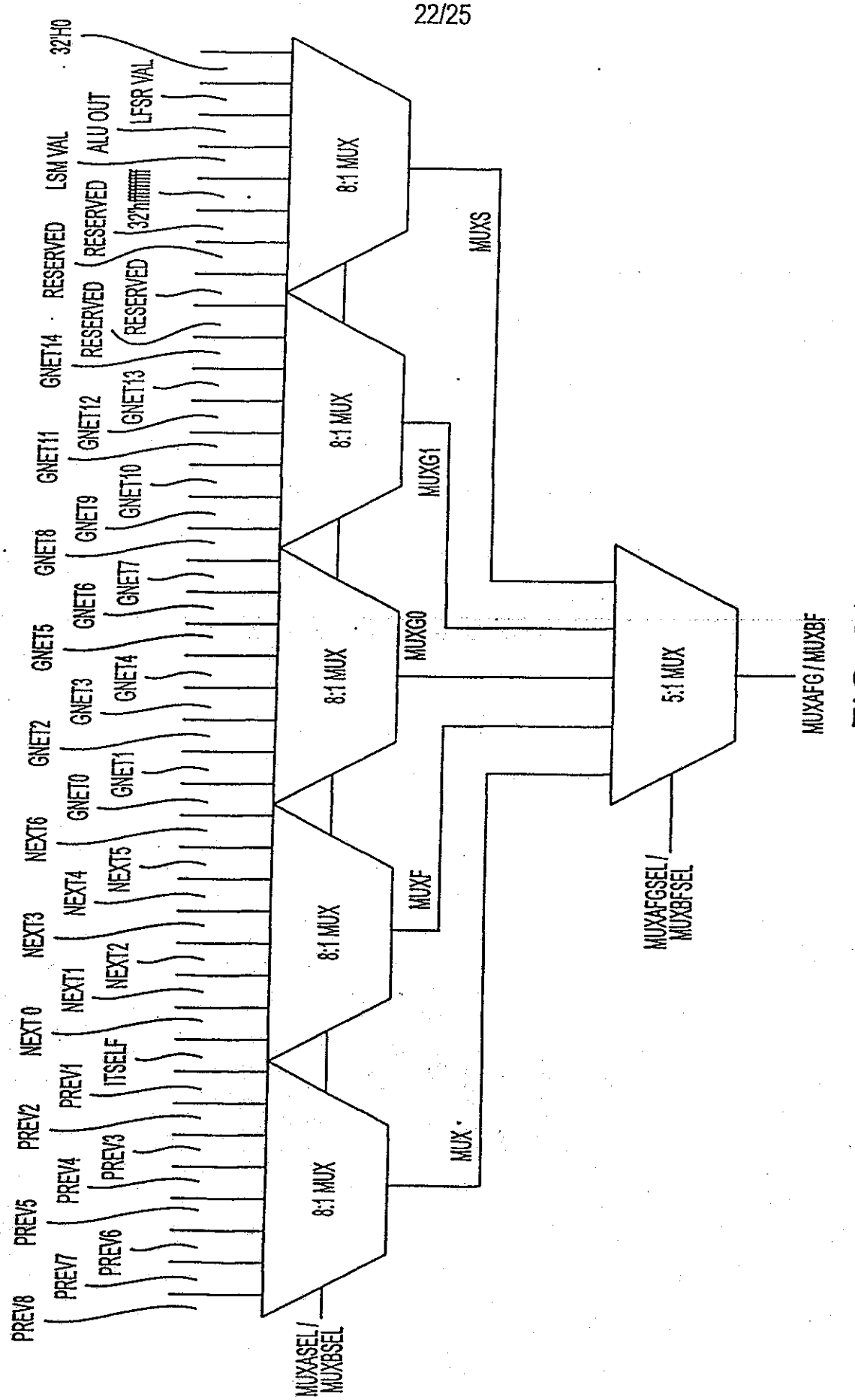

PREV8
PREV7
PREV6
PREV5
PREV4
PREV3
PREV2
PREV1
ITSELF
NEXT0
NEXT1
NEXT2
NEXT3
NEXT4
NEXT5
NEXT6
GNET0
GNET1
GNET2
GNET3
GNET4
GNET5
GNET6
GNET7
GNET8
GNET9
GNET10
GNET11
GNET12
GNET13
GNET14
RESERVED
RESERVED
RESERVED
LSM VAL
32'h0
LFSR VAL
ALU OUT
32'hffffffff
8:1 MUX
8:1 MUX
8:1 MUX
8:1 MUX
5:1 MUX
MUXS
MUXG1
MUXG0
MUXF
MUX
MUXASEL / MUXBSEL
MUXAFGSEL / MUXBFSEL
MUXAFG / MUXBF

FIG. 21

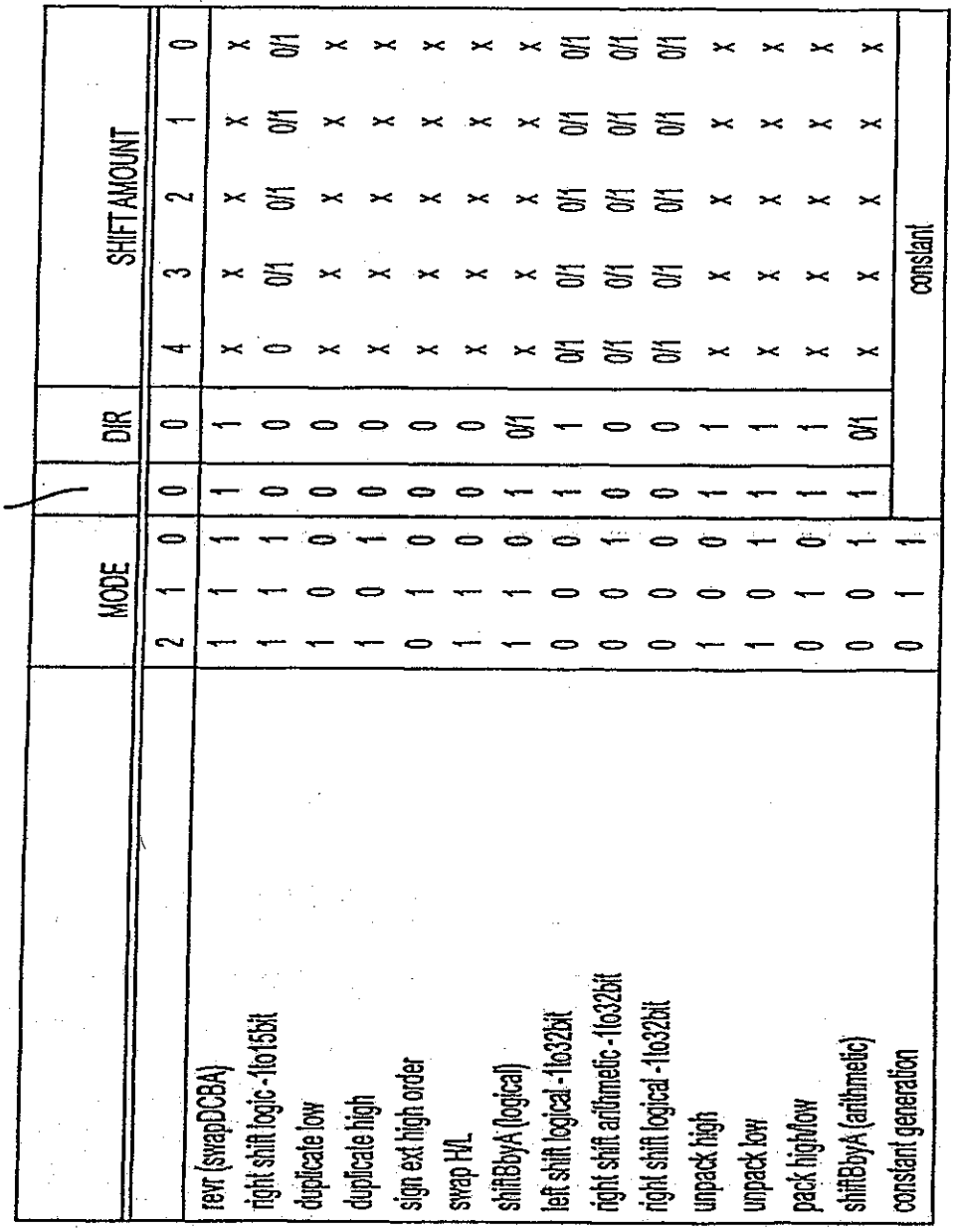

SHIFTDIR
MODE
DIR
SHIFT AMOUNT
2 1 0
0
4 3 2 1 0
revr (swapDCBA)
right shift logic -1to15bit
duplicate low
duplicate high
sign ext high order
swap H/L
shiftBbyA (logical)
left shift logical -1to32bit
right shift arithmetic -1to32bit
right shift logical -1to32bit
unpack high
unpack low
pack high/low
shiftBbyA (arithmetic)
constant generation
constant

FIG. 22

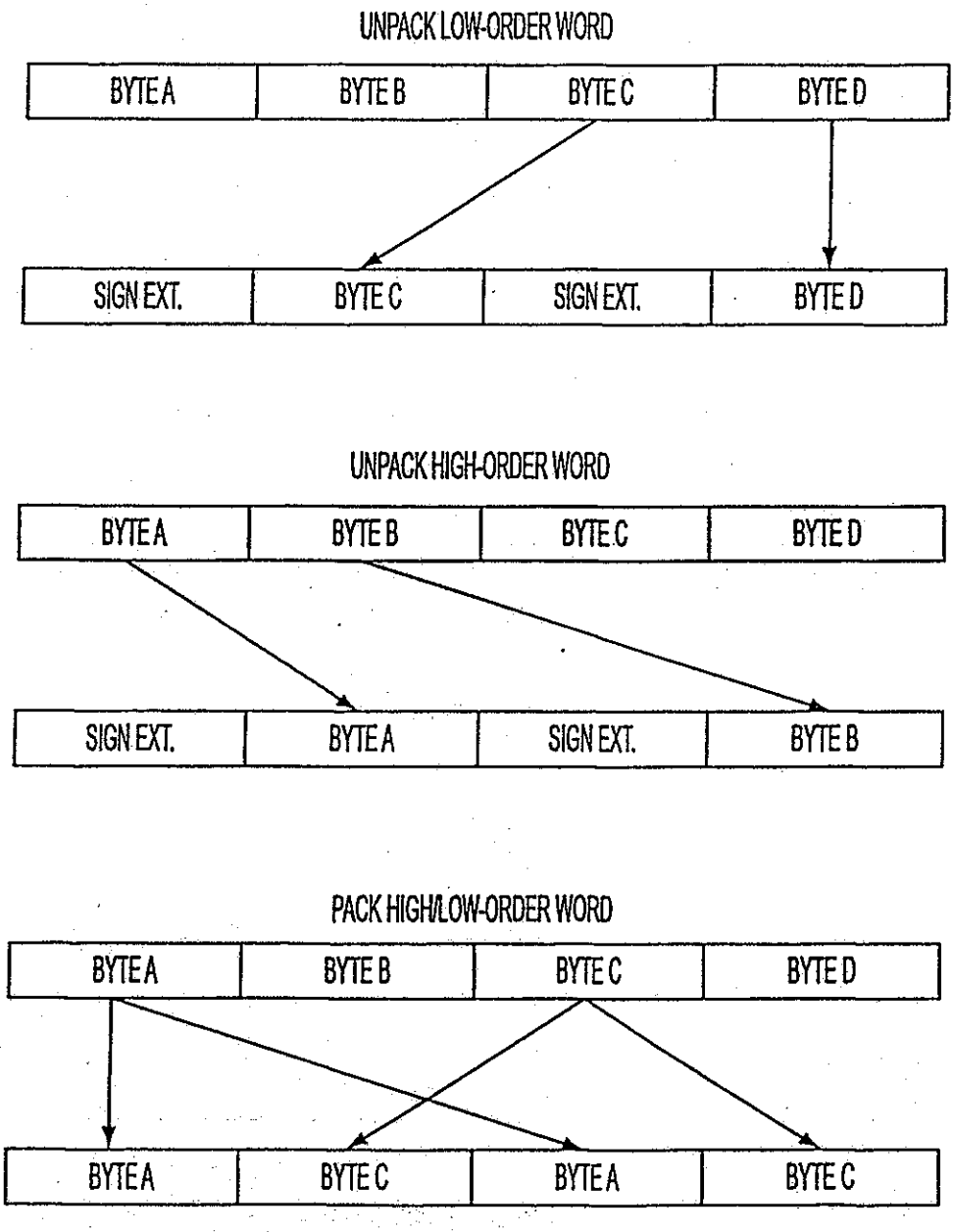

UNPACK LOW-ORDER WORD
BYTE A
BYTE B
BYTE C
BYTE D
SIGN EXT.
BYTE C
SIGN EXT.
BYTE D
UNPACK HIGH-ORDER WORD
BYTE A
BYTE B
BYTE C
BYTE D
SIGN EXT.
BYTE A
SIGN EXT.
BYTE B
PACK HIGH/LOW-ORDER WORD
BYTE A
BYTE B
BYTE C
BYTE D
BYTE A
BYTE C
BYTE A
BYTE C

FIG. 23

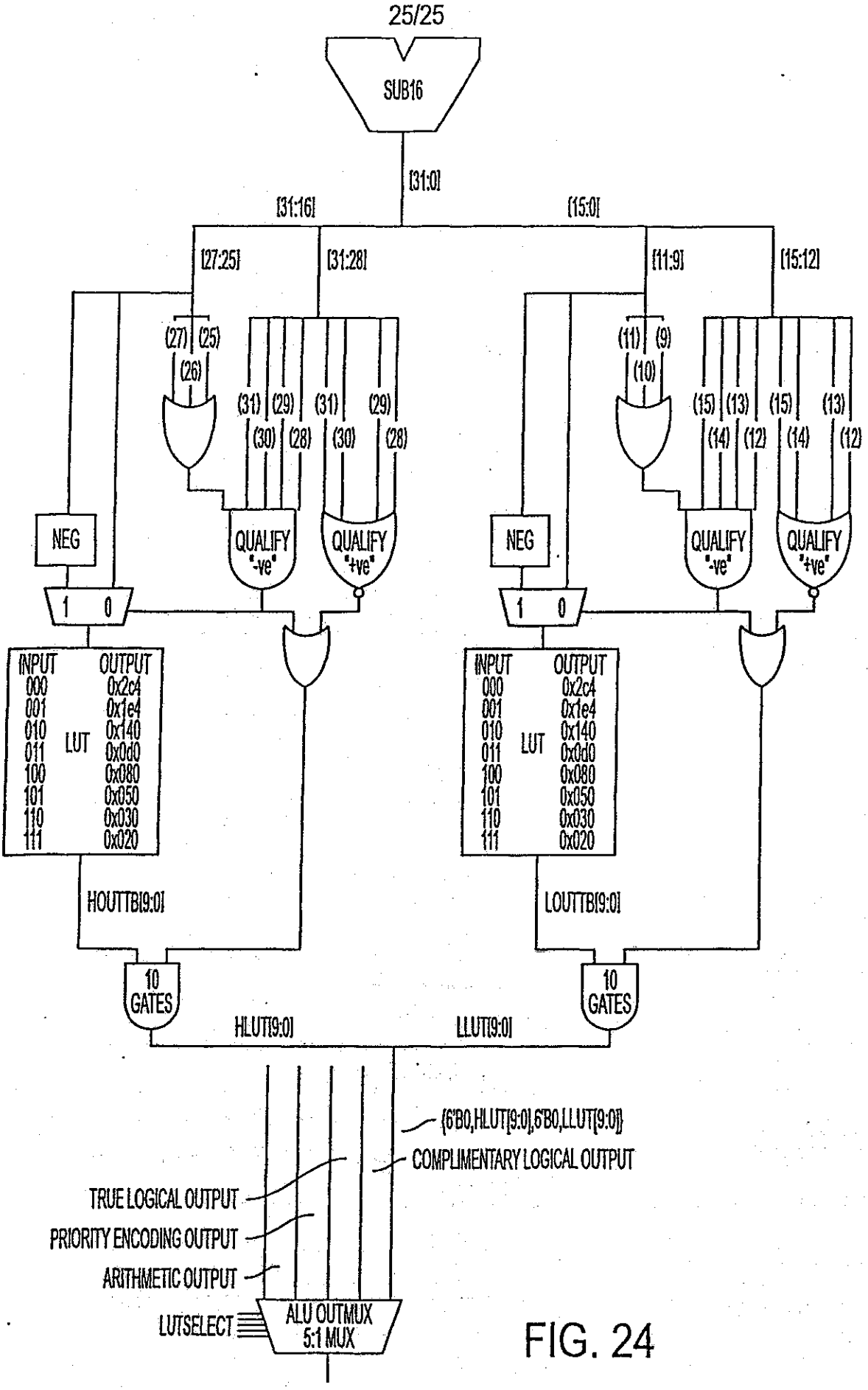

SUB16
[31:0]
[31:16]
[15:0]
[27:25]
[31:28]
[11:9]
[15:12]
(27) (25)
(26)
(31) (29)
(30) (28)
(31) (29)
(30) (28)
(11) (9)
(10)
(15) (13)
(14) (12)
(15) (13)
(14) (12)
NEG
QUALIFY "-ve"
QUALIFY "+ve"
NEG
QUALIFY "-ve"
QUALIFY "+ve"
1 0
1 0
INPUT OUTPUT
000 0x2c4
001 0x1e4
010 0x140
011 0x0d0
100 0x080
101 0x050
110 0x030
111 0x020
LUT
INPUT OUTPUT
000 0x2c4
001 0x1e4
010 0x140
011 0x0d0
100 0x080
101 0x050
110 0x030
111 0x020
LUT
HOUTTB[9:0]
LOUTTB[9:0]
10 GATES
10 GATES
HLUT[9:0]
LLUT[9:0]
{6'B0,HLUT[9:0],6'B0,LLUT[9:0]}
COMPLIMENTARY LOGICAL OUTPUT
TRUE LOGICAL OUTPUT
PRIORITY ENCODING OUTPUT
ARITHMETIC OUTPUT
LUTSELECT
ALU OUTMUX 5:1 MUX

FIG. 24