PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :		(11) International Publication Number:	WO 96/21186
G06F 9/38	A2	(43) International Publication Date:	11 July 1996 (11.07.96)

(21) International Application Number: PCT/IB95/01013

(22) International Filing Date: 16 November 1995 (16.11.95)

(30) Priority Data:
08/366,958 30 December 1994 (30.12.94) US

(71) Applicant: PHILIPS ELECTRONICS N.V. [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).

(71) Applicant *(for SE only)*: PHILIPS NORDEN AB [SE/SE]; Kottbygatan 5, Kista, S-164 85 Stockholm (SE).

(72) Inventors: MEHRA, Vijay, K.; 5388 Shamrock Common, Freemont, CA 94555 (US). SLAVENBURG, Gerrit, Ary; 304 Langton Avenue, Los Altos, CA 94022 (US).

(74) Agent: DE HAAS, Laurens, J.; Internationaal Octrooibureau B.V., P.O. Box 220, NL-5600 AE Eindhoven (NL).

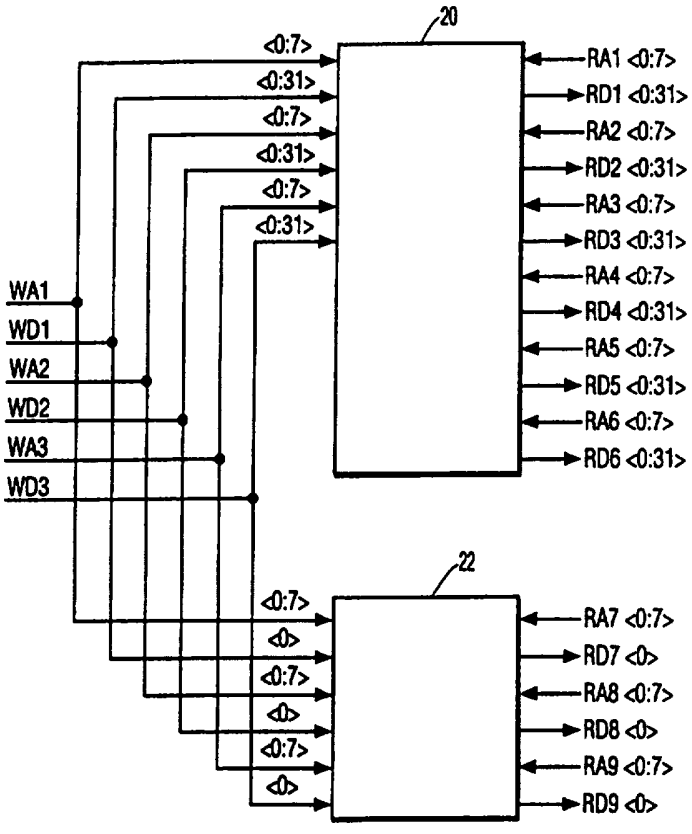
(81) Designated States: JP, KR, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: PLURAL MULTIPORT REGISTER FILE TO ACCOMMODATE DATA OF DIFFERING LENGTHS

(57) Abstract

A multiport register file includes a first file unit having registers of a first width and a second file unit having registers of a second width. The second width being less than the first width. Data is written to both the first and second file unit in one write operation and independently readable separately from the first and second file units. The first file unit accommodates data destined to be operands for functional units of a VLIW processor, or result data from those functional units. The second file unit accommodates guard bits for conditioning operation of those functional units.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	**GB**	United Kingdom	**MW**	Malawi
AT	Austria	**GE**	Georgia	**MX**	Mexico
AU	Australia	**GN**	Guinea	**NE**	Niger
BB	Barbados	**GR**	Greece	**NL**	Netherlands
BE	Belgium	**HU**	Hungary	**NO**	Norway
BF	Burkina Faso	**IE**	Ireland	**NZ**	New Zealand
BG	Bulgaria	**IT**	Italy	**PL**	Poland
BJ	Benin	**JP**	Japan	**PT**	Portugal
BR	Brazil	**KE**	Kenya	**RO**	Romania
BY	Belarus	**KG**	Kyrgystan	**RU**	Russian Federation
CA	Canada	**KP**	Democratic People's Republic of Korea	**SD**	Sudan
CF	Central African Republic			**SE**	Sweden
CG	Congo	**KR**	Republic of Korea	**SG**	Singapore
CH	Switzerland	**KZ**	Kazakhstan	**SI**	Slovenia
CI	Côte d'Ivoire	**LI**	Liechtenstein	**SK**	Slovakia
CM	Cameroon	**LK**	Sri Lanka	**SN**	Senegal
CN	China	**LR**	Liberia	**SZ**	Swaziland
CS	Czechoslovakia	**LT**	Lithuania	**TD**	Chad
CZ	Czech Republic	**LU**	Luxembourg	**TG**	Togo
DE	Germany	**LV**	Latvia	**TJ**	Tajikistan
DK	Denmark	**MC**	Monaco	**TT**	Trinidad and Tobago
EE	Estonia	**MD**	Republic of Moldova	**UA**	Ukraine
ES	Spain	**MG**	Madagascar	**UG**	Uganda
FI	Finland	**ML**	Mali	**US**	United States of America
FR	France	**MN**	Mongolia	**UZ**	Uzbekistan
GA	Gabon	**MR**	Mauritania	**VN**	Viet Nam

Plural multiport register file to accommodate data of differing lengths.

BACKGROUND OF THE INVENTION

1. Field of the invention

The invention relates to a processor comprising

- a plurality of functional units;
- a register file containing
  - a first number of addressable first registers, each having a first number of bits;
  - a second number of addressable second registers, each having a second number of bits smaller than the first number of bits;
  - a plurality of write ports, each having an associated write address port, the functional units being coupled to respective write ports and associated write address ports,
  - a plurality of read ports, each having an associated read address port, the functional units being coupled to respective read ports and associated read address ports.

The invention also relates to a register file for use in such a processor.

2. Related art

Multiport register files are used for digital data processors which need to access plural registers simultaneously. In particular, such register files are useful for VLIW (Very Long Instruction Word) processors. Such processors also include an instruction register accommodating plural operation codes and a plurality of functional units for executing the plural operations codes, starting simultaneously in a single machine cycle.

Multiport register files can be used in other types of processors as well.

A prior art multiport register file is shown in Fig. 1. This file includes 128 32 bit registers.

To the left of the file are shown write address ports, WA1, WA2, and WA3, each being eight bits wide. Also shown on the left are write ports WD1, WD2, WD3, each being 32 bits wide. Results from 3 functional units can be written simultaneously on the

write ports at the addresses specified on the write address ports.

To the right of the file are shown read address ports RA1, RA2, RA3, RA4, RA5, RA6, RA7, RA8, and RA9, each being eight bits wide. Also shown on the right are read ports, RD1, RD2, RD3, RD4, RD5, RD6, RD7, RD8, and RD9, each being 32 bits wide. Up to nine operands destined for the functional units can be read from this file simultaneously on the write ports from the addresses specified on the write address ports.

In VLIW processors, guard bits are used to condition writing of results from the functional units to the multiport register file. Guard bits become necessary in VLIW processors because of branching delays, as explained in EP 479 390 (PHA 1209). The functional units execute operations during a branch delay before the processor resolves whether results of those operations will actually be used. After the operations are completed, each functional unit will write results to the register file only if an associated guard bit has an appropriate value.

There are nine read ports in this particular file unit, because the VLIW processor in question has an instruction word accommodating 3 operations. Each operation will typically require two data operands and a guard bit. There are three write ports to accommodate a result from each of 3 simultaneously executing functional units. Each read or write port has an associated address port.

Ordinarily, the guard bits are to be supplied from the multiport register file. Guard bits, or multibit guard values, are generally much smaller than the thirty-two bit registers and thirty-two bit read and write ports available in the prior art register file. Where the writing from each functional unit is to be conditioned by a guard bit or value, a great deal of unnecessary circuitry is necessary, in particular extra 32-bit write and read ports and extra 8-bit write and read address ports.

SUMMARY OF THE INVENTION

The object of the invention is to reduce circuitry necessary for operation of the processor.

This object is achieved because the processor is characterized in that

- the first registers are disposed in a first file unit along with associated ones of the write ports, write address ports, read ports and read address ports; and
- the second registers are disposed in a second file unit along with associated ones of the write ports, write address ports, read ports, and read address ports.

A guard bit will be stored in the second registers for VLIW processors. For

other types of processors, other types of short data can be stored in the second file unit. Such short data can include flags, for example.

BRIEF DESCRIPTION OF THE DRAWING

The invention will now be explained by way of non-limitative example with reference to the following figures:

Fig. 1 shows a prior art multi-port register file.

Fig. 2 shows a multi-port register file according to the invention.

Fig. 3 shows a floor plan of a register file.

Fig. 4 shows a register cell which would be suited to use in the prior art floor plan.

Fig. 5 shows a floor plan of register files in accordance with the invention.

Fig. 6 shows a register file cell suited for use in the data portion of the register files of Fig. 5.

Fig. 7 shows a register file cell suited for use in the guard portion of the register files of Fig. 5.

Fig. 8 shows a decoder for converting read and write address signals into read and write enable signals.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Fig. 2 shows a multi-port register file according to the invention. The file is divided into two pieces, a data file unit 20, which is again a 128 register, 32-bit wide, file, and a guard file unit 22, which is 128 register, 1-bit wide file.

The write address inputs WA1, WA2, WA3 and write data inputs WD1, WD2, and WD3 are as indicated in the prior art. However, only one bit of the data inputs need be routed to the guard file 22. No routing circuits are necessary because the guard bits are written to both files and only read from one. The read address inputs RA1, RA2, RA3, RA4, RA5, and RA6 and the read data outputs RD1, RD2, RD3, RD4, RD5, and RD6 are dedicated to the data file 20. The read address inputs RA7, RA8, and RA9 and the read data outputs RD7, RD8, and RD9 are dedicated to the guard file 22. Since the read data outputs dedicated to the guard file are only one bit wide, substantial circuitry is saved over the prior art implementation in which 3 extra 32-bit wide data buses are necessary at the output of the data file 20. The read ports and write ports fit into the architecture of European Patent Application No. 605 927 (PHA 21.777) just as the prior art multiport register file did.

Fig. 3 shows a floor plan of a register file in accordance with Fig. 1. The file consists of a matrix of register cells, arranged in rows and columns. For brevity only the top and bottom rows and left and right columns are shown. There are thirty-two columns, one for each bit of the registers. There are 128 rows, one for each of the registers.

Fig. 4 shows a register cell suitable for use in the floor plan of Fig. 3. At the left are input respective bits of the write data signals WD1, WD2, and WD3, which bits are connected to MOSFETs 401, 402, and 403, respectively. The gates of MOSFETs 401, 402, and 403 are coupled with respective bits of the write enable signals WE1, WE2, and WE3. Junction 425 functions as a wired OR inputting to inverter 423. A feedback inverter 424 is coupled between the input and output of inverter 423. The output 404 of inverter 423 is coupled to the gates of MOSFETs 405-413. MOSFETs 405-413 are connected to respective bits of RD1-RD9. MOSFETs 405-413 are also connected to MOSFETS 414-422, respectively. The gates of MOSFETs 414-422 are connected to respective bits of the read enable signals RE1-RE9, respectively.

Thus in the register file of Fig. 3, one could expect the following:

Location and type of component	Number of components
write data wires per cell	3
read data wires per cell	9
write enable wires per cell	3
read enable wires per cell	9
horizontal wires (i.e. read enable and write enable) per row	12
vertical wires (i.e. read data and write data) per column	12
number of transistors per cell	25
total horizontal wires in register file core	12*128 = 1536
total vertical wires in register file core	12*32 = 384
total transistors in register file core	25*32*128 = 102400

Fig. 5 shows the floor plan of the multiport register file in accordance with the invention. The new multiport register file includes a data register file unit (a) having the same floor plan as the prior art and a guard register file unit (b) having one column of 128 register cells. Although the floor plan of the data register file unit (a) is the same as for the prior art register file, the register cells needed for the new data register file are vastly simplified. The cells needed for the guard register file unit (b) are simpler still.

Fig. 6 shows a register cell which would function in the data register file unit (a) according to the invention. The left portion of the register cell is the same as that in Fig. 4, with like components having like reference numerals. However the right portion of the

cell is simplified with transistors 605-610 being substituted for 405-413 and transistors 614-619 being substituted for transistors 414-422. In other words the cell of Fig. 6 has 6 less transistors and correspondingly less read lines than the cell of Fig. 4.

Fig. 7 shows a register cell which would function in the guard register unit (b) of Fig. 5. The left portion of this cell resembles the left portion of the cell of Fig. 4. However, the right portion is even more simplified than the cell of Fig. 6. MOSFETs 705-707 are substituted for MOSFETs 405-413 and MOSFETs 714-716 are substituted for MOSFETs 414-412. MOSFETs 705-707 are connected to respective bits of read data lines RD7-RD9. MOSFETs 714-716 are coupled with respective bits of read enable lines RE7-RE9. In other words the cells of Fig. 7 have 12 less transistors than the cells of Fig. 4 and 6 less transistors than the cells of Fig. 6, with correspondingly fewer read lines. Since the read enable lines RE7-9 are not needed in the data register file unit (a) and the read enable lines RE1-6 lines are not needed in the guard register file unit (b), the lines RE7-9 can occupy the same horizontal spaces as allocated to three of the lines RE1-6. Accordingly, no additional horizontal wire space is needed for RE7-9.

Thus in the register file of Fig. 5, one finds the following:

Type and location of component	Number of components
Write data wires per data cell	3
Read data wires per data cell	6
Write enable wires per data cell	3
Read enable wires per data cell	6
Horizontal wires (i.e. read and write enable) per data row	9
Vertical wires (i.e. read and write data) per data column	9
Transistors per data cell	19
Write data wires per guard cell	3
Read data wires per guard cell	3
Write enable wires per guard cell	3
Read enable wires per guard cell	3
Horizontal wires (read enable and write enable) per guard row	6
Vertical wires (read data and write data) per guard column	6
Transistors per guard cell	13
Total horizontal wire spaces	9*128= 1152
Total vertical wires	9*32+6*1= 294
Total transistors	19*32*128 + 13*1*128 = 79488

Thus the embodiment of Figs. 2 and 5-7 has 22,912 fewer transistors, 384 fewer horizontal wire spaces and 90 fewer vertical wires than the embodiment of Figs. 1 and 3-4.

The read enable inputs and write enable inputs in figures 4, 6, and 7 are obtained from the read address inputs and the write address inputs using a decoder circuit as shown in Fig. 8. Standard address decoding blocks are shown at 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812. These blocks convert the eight bit addresses WA1, WA2, WA3, RA1, RA2, RA3, RA4, RA5, RA6, RA7, RA8, RA9 into 128 bit enable signals WE1, WE2, WE3, RE1, RE2, RE3, RE4, RE5, RE6, RE7, RE8, and RE9, respectively.

Each bit of the write enable signals relates to a respective row of the register file, and goes to each cell in that respective row of both the data register file unit (a) and the guard register file unit (b). Each bit of the read enable signals RE1, RE2, RE3, RE4, RE5, and RE 6 relates to a respective row of the data register file unit (a) and goes to each cell in that respective row of the data register file unit (a). Each bit of the read enable signals RE7, RE8, and RE9 relates to a respective row of the guard register file unit (b) and goes to each cell in that respective row of the guard register file unit (b). Thus, for example, bit 1 of WE1 goes to each cell in row 1 of both register file units; bit 1 of RE1 goes to each cell in row 1 of the data register file unit (a); bit 1 of RE7 goes to the cell in row 1 of the guard file register unit (b); and so forth.

Those of ordinary skill in the art will readily recognize that multi-port register files according to the invention can have a variety of other embodiments. These embodiments include the following. The data registers can be any width, such as 16 bits, which is used for operand or result data in a particular processor. The guard registers can be slightly wider if multibit guard or flag values are to be used. More register files may be used if the processor needs to use data of other, different widths.

CLAIMS:

1. A processor comprising

- a plurality of functional units;
- a register file containing
  - a first number of addressable first registers, each having a first number of bits;
  - a second number of addressable second registers, each having a second number of bits smaller than the first number of bits;
  - a plurality of write ports, each having an associated write address port, the functional units being coupled to respective write ports and associated write address ports,
  - a plurality of read ports, each having an associated read address port, the functional units being coupled to respective read ports and associated read address ports,

characterized, in that the processor is capable of supplying a write address which addresses both a particular first register and a particular second register to the write address port associated with at least one of the write ports, the register file being arranged for writing data on the at least one of the write ports both to the particular first register and to the particular second register upon receiving said write address on the associated write address port, the particular first register and the particular second register being independently addressable via the read address ports.

2. The processor of claim 1 wherein

- the first registers are disposed in a first file unit along with associated ones of the write ports, write address ports, read ports and read address ports; and
- the second registers are disposed in a second file unit along with associated ones of the write ports, write address ports, read ports, and read address ports.

3. The processor of claim 1 or 2 wherein the register file is a multiport register file, at least one of the first and/or second registers being accessible from more than one of the read ports.

4. The processor of Claim 1, 2 or 3 wherein at least one of the functional

units is arranged for writing a result of an operation conditionally, dependent on a guard bit read from the particular second register.

5. The processor of Claim 1, 2, 3 or 4 wherein the second number of bits is one.

6. The processor of any one of Claims 1 to 5, wherein the processor is a VLIW processor including an instruction register accommodating plural operation codes for execution in parallel by the functional units, starting simultaneously in a single machine cycle.

7. A processor comprising

- a plurality of functional units;
- a register file containing
  - a first number of addressable first registers, each having a first number of bits;
  - a second number of addressable second registers, each having a second number of bits smaller than the first number of bits;
  - a plurality of write ports, each having an associated write address port, the functional units being coupled to respective write ports and associated write address ports,
  - a plurality of read ports, each having an associated read address port, the functional units being coupled to respective read ports and associated read address ports,

characterized, in that the register file is a multiport register file, at least one of the first and/or second registers being accessible from more than one of the read ports.

8. The processor of claim 7 wherein

- the first registers are disposed in a first file unit along with associated ones of the write ports, write address ports, read ports and read address ports; and
- the second registers are disposed in a second file unit along with associated ones of the write ports, write address ports, read ports, and read address ports.

9. A multiport register file suitable for use in a processor according to any one of the claims 1 to 8.

10. A method of operating a processor, comprising the steps of

- supplying data and a write address from a first functional unit respectively to a write port and a corresponding write address port of a register file;
- storing said data in a first register addressed by said write address and storing only part

of said data in a second register also addressed by said write address;

- using said data and/or the part of said data in either one or both of a first and a second access operation,

the first access operation comprising

- supplying a read address of the second register to a read address port of the register file,
- reading the part of said data from the second register into a second functional unit,
- conditioning writing of results from said second functional unit upon a value of the part;

the second access operation comprising

- supplying a read address of the first register to a read address port of the register file,
- reading said data from the first register into a third functional unit,
- operating on said data with said third functional unit.

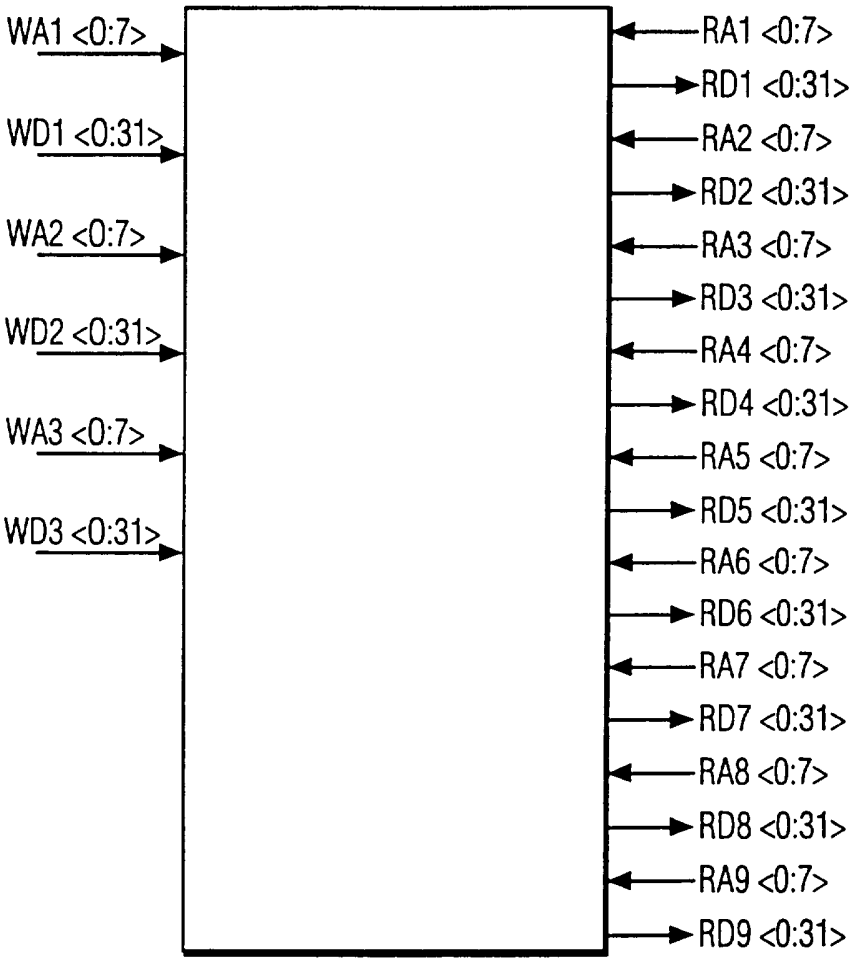

WA1 <0:7>
WD1 <0:31>
WA2 <0:7>
WD2 <0:31>
WA3 <0:7>
WD3 <0:31>
RA1 <0:7>
RD1 <0:31>
RA2 <0:7>
RD2 <0:31>
RA3 <0:7>
RD3 <0:31>
RA4 <0:7>
RD4 <0:31>
RA5 <0:7>
RD5 <0:31>
RA6 <0:7>
RD6 <0:31>
RA7 <0:7>
RD7 <0:31>
RA8 <0:7>
RD8 <0:31>
RA9 <0:7>
RD9 <0:31>

FIG.1

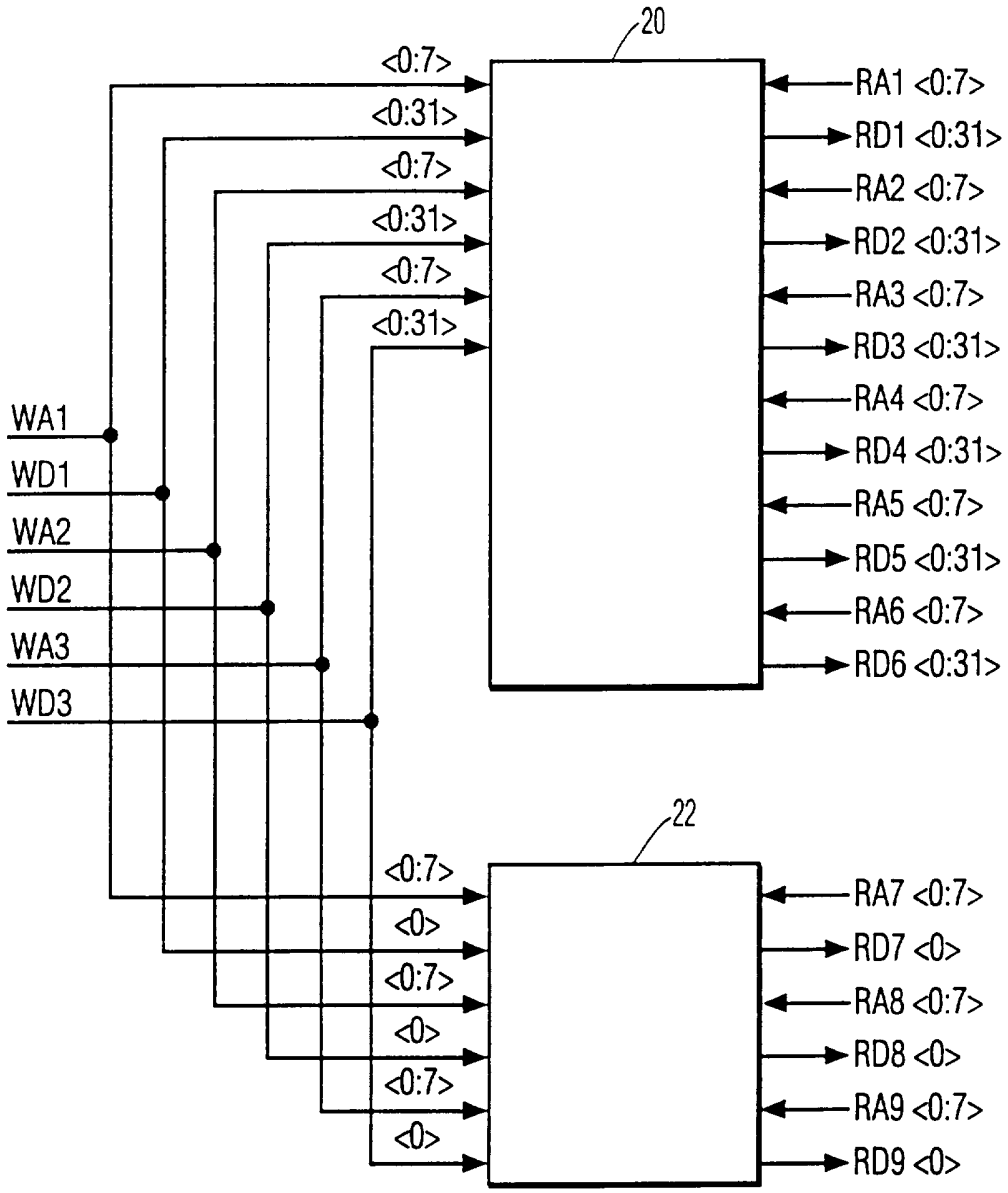

20
<0:7>
<0:31>
<0:7>
<0:31>
<0:7>
<0:31>
WA1
WD1
WA2
WD2
WA3
WD3
RA1 <0:7>
RD1 <0:31>
RA2 <0:7>
RD2 <0:31>
RA3 <0:7>
RD3 <0:31>
RA4 <0:7>
RD4 <0:31>
RA5 <0:7>
RD5 <0:31>
RA6 <0:7>
RD6 <0:31>
22
<0:7>
<0>
<0:7>
<0>
<0:7>
<0>
RA7 <0:7>
RD7 <0>
RA8 <0:7>
RD8 <0>
RA9 <0:7>
RD9 <0>

FIG.2

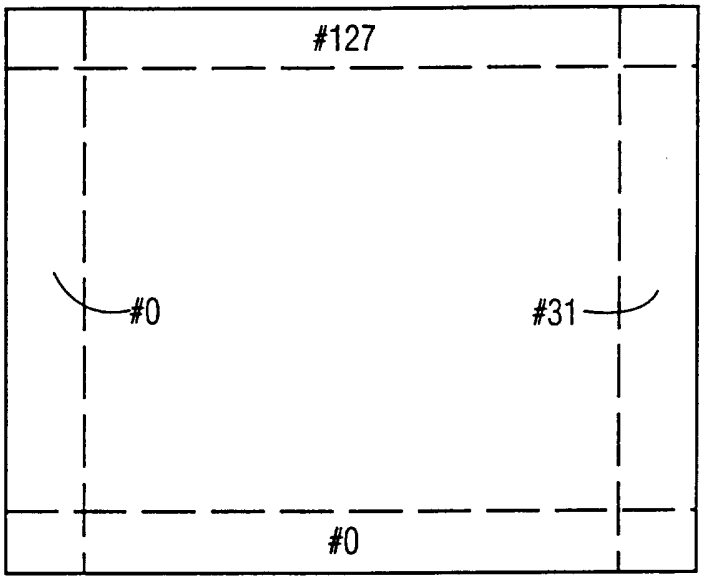

#127
#0
#31
#0

FIG. 3

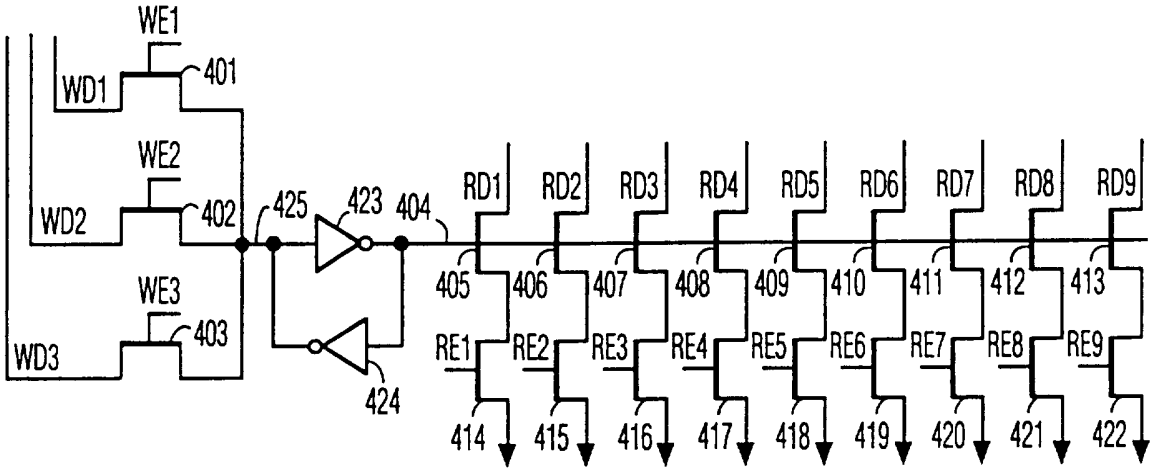

WE1
WD1
401
WE2
WD2
402
425
423
404
RD1
RD2
RD3
RD4
RD5
RD6
RD7
RD8
RD9
WE3
WD3
403
424
405
406
407
408
409
410
411
412
413
RE1
RE2
RE3
RE4
RE5
RE6
RE7
RE8
RE9
414
415
416
417
418
419
420
421
422

FIG. 4

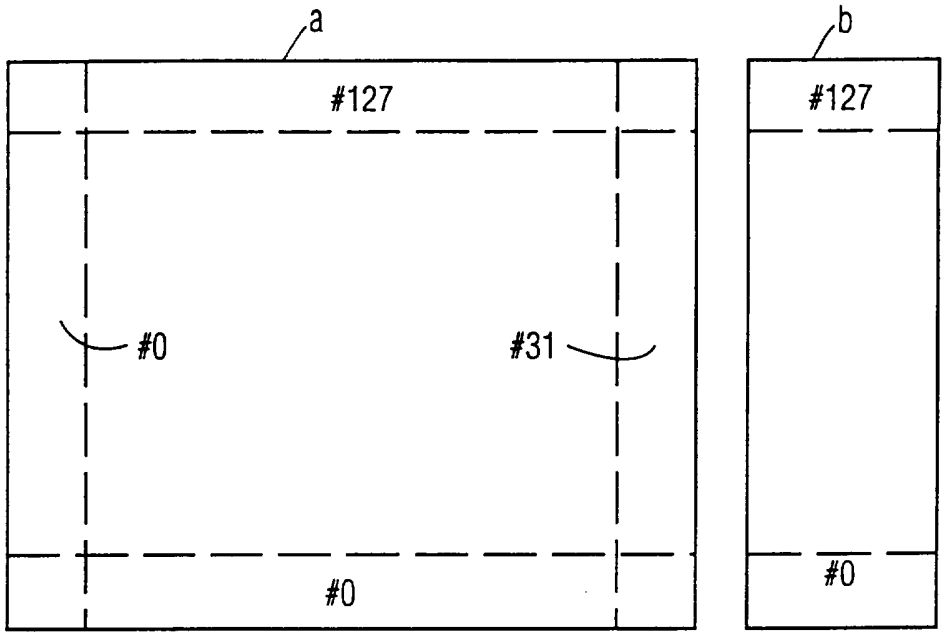


FIG. 5

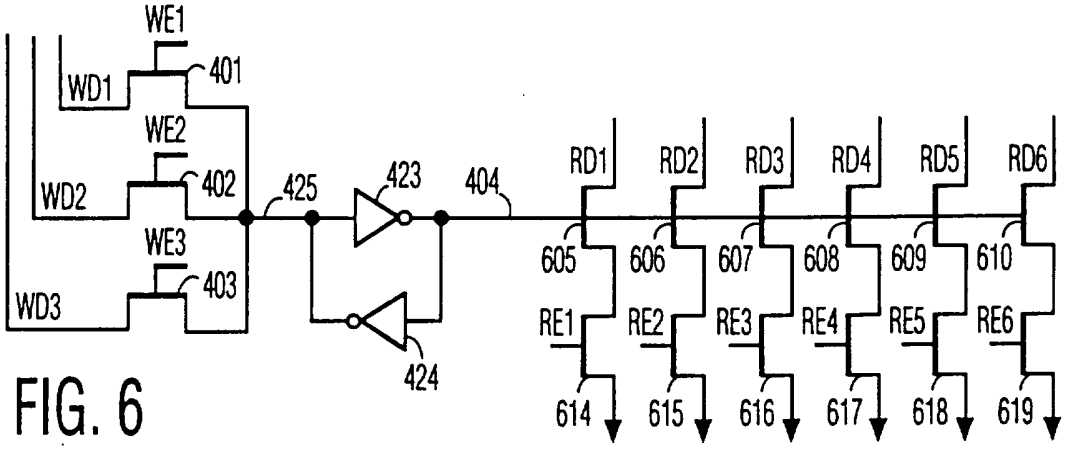


FIG. 6

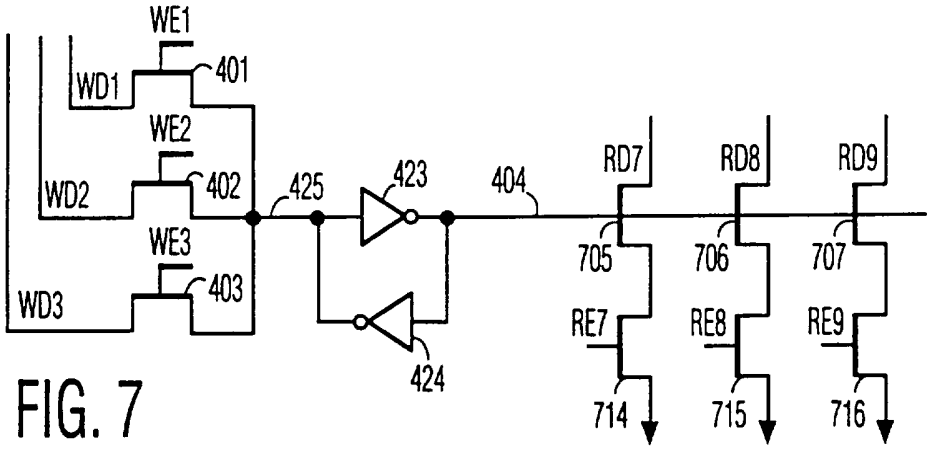


FIG. 7

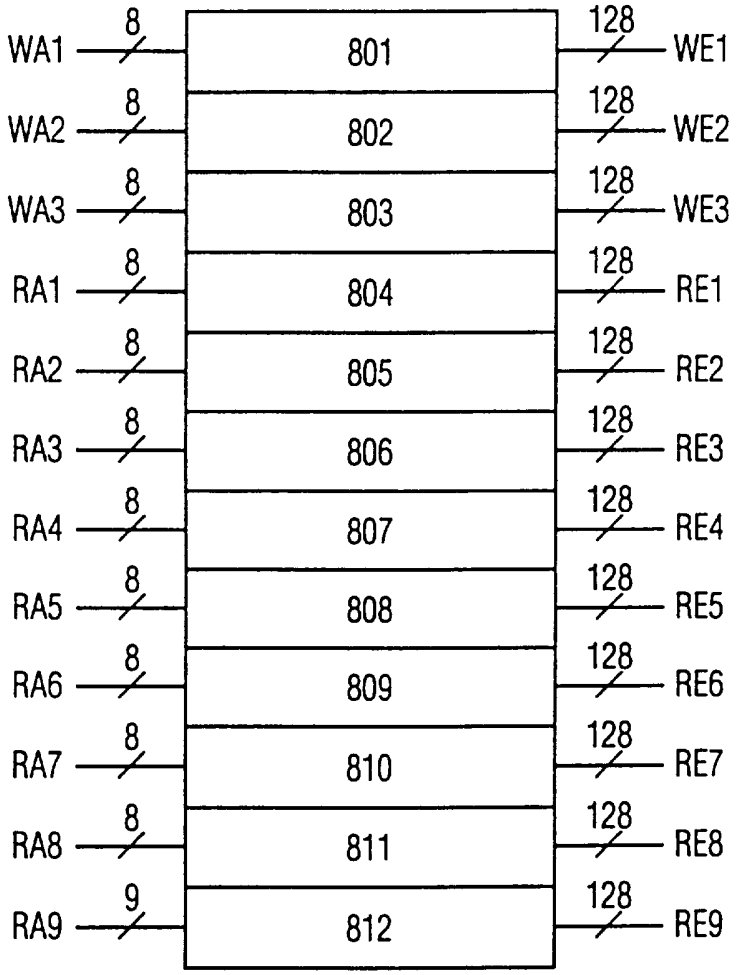

WA1
8
WA2
8
WA3
8
RA1
8
RA2
8
RA3
8
RA4
8
RA5
8
RA6
8
RA7
8
RA8
8
RA9
9
801
802
803
804
805
806
807
808
809
810
811
812
128
WE1
128
WE2
128
WE3
128
RE1
128
RE2
128
RE3
128
RE4
128
RE5
128
RE6
128
RE7
128
RE8
128
RE9

FIG. 8