(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 January 2007 (04.01.2007)

PCT

(10) International Publication Number
WO 2007/002324 A2

(51) International Patent Classification:
*G11C 29/00* (2006.01)

(21) International Application Number:
PCT/US2006/024360

(22) International Filing Date: 23 June 2006 (23.06.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/693,631 24 June 2005 (24.06.2005) US

(71) Applicant *(for all designated States except US)*: **METARAM, INC.** [US/US]; 181 Metro Drive, San Jose, CA 95110 (US).

(71) Applicant and
(72) Inventor: **RAJAN, Suresh, N.** [US/US]; 3284 Isadora Drive, San Jose, CA 95132 (US).

(74) Agent: **STATTLER, John**; Stattler Johansen & Adeli LLP, P.O. Box 51860, Palo Alto, CA 94303-0728 (US).

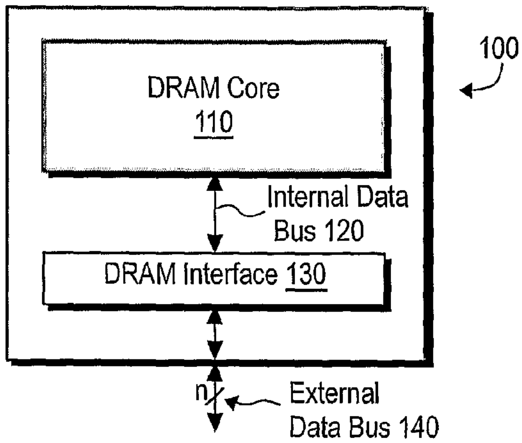
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Published:**
— *without international search report and to be republished upon receipt of that report*

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

**(54) Title:** AN INTEGRATED MEMORY CORE AND MEMORY INTERFACE CIRCUIT




**(57) Abstract:** A memory device comprises a first and second integrated circuit dies. The first integrated circuit die comprises a memory core as well as a first interface circuit. The first interface circuit permits full access to the memory cells (*e.g.*, reading, writing, activating, pre-charging and refreshing operations to the memory cells). The second integrated circuit die comprises a second interface that interfaces the memory core, via the first interface circuit, an external bus, such as a synchronous interface to an external bus. A technique combines memory core integrated circuit dies with interface integrated circuit dies to configure a memory device. A speed test on the memory core integrated circuit dies is conducted, and the interface integrated circuit die is electrically coupled to the memory core integrated circuit die based on the speed of the memory core integrated circuit die.

WO 2007/002324 A2

# AN INTEGRATED MEMORY CORE AND MEMORY INTERFACE CIRCUIT

Related Applications:

This patent application claims the benefit to United States Provisional Patent Application entitled "Methods and Apparatus for Integrating Multi-Chip Memory Devices," Serial number 60/693,631, filed on June 24, 2005.

## BACKGROUND OF THE INVENTION

Field of the Invention:

The present invention is directed toward the field of building custom memory systems cost-effectively for a wide range of markets.

Art Background:

Dynamic Random Access Memory (DRAM) is the most popular type of volatile memory and is widely used in a number of different markets. The popularity of DRAMs is mostly due to their cost-effectiveness (Mb/$). The PC main memory market has traditionally been the largest consumer of DRAMs. However, in recent times, other important markets have adopted DRAMs. A report published by De Dios and Associates showed that in 2004, the PC main memory market consumed only 50% of the total DRAM bits.

Several of the non-PC markets use specialty or legacy memories. Specialty memory is typically memory that is not used by the PC main memory but is memory that is designed for one or more niche markets. For example, the PC graphics market uses GDDR (Graphics Dual Data Rate) DRAM. Similarly, some segments of the network infrastructure market use FCRAM (Fast Cycle RAM) or RLDRAM (Reduced Latency DRAM). Legacy memory is typically memory that was used in the past but is not used in that particular market segment now. For example, SDRAM (Synchronous DRAM) was used for PC main memory from ~1997 to ~2001 but is no longer used today for PC main memory. Instead, most cellular phones and handheld (or mobile) devices use SDRAM today.

Bringing a new DRAM architecture into the market requires significant investment of time and money. For example, it typically takes 4 years for JEDEC to approve a new DRAM architecture. DRAM makers must then spend hundreds of millions of dollars to productize the new architecture. Unless the investment is amortized over an extremely large number of devices, the cost of the new devices will be high. In addition, the DRAM makers have optimized their manufacturing flow for high volumes. Any deviation from the norm disrupts the flow. This is the reason why specialty and legacy memory typically carry a price premium over memory used by the PC main memory market (which is usually referred to as commodity memory).

Given the time and money required to bring a new DRAM architecture into the market, it is obvious that the industry does not have the luxury of being able to define a DRAM architecture that exclusively meets the needs of the smaller markets for DRAMs. For example, it is difficult for the DRAM makers to cost-effectively produce a DRAM that perfectly meets the needs of the cell phone market. Therefore, it is even more difficult for each cell phone maker (e.g. Nokia or Motorola) to design DRAMs tailor made for its phones. So, cell phone designers are forced to choose the DRAM architecture that is least objectionable from their perspective. Hence the selection of SDRAM for cell phones.

This situation will become even worse in the future. Most analyst projections show that not only will DRAM usage expand rapidly into newer markets but also that the DRAM bit consumption growth rate will be higher for non-PC markets. The needs of these markets are quite different from the needs of the PC main memory market. Clearly, there is a need in the market for a way to quickly and cost-effectively build custom memory that is tailor made for a customer's exact needs.

## SUMMARY OF THE INVENTION

A memory device comprises a first integrated circuit die. The first integrated circuit die comprises a memory core, with a plurality of memory cells, and a first interface circuit for accessing the memory cells of the memory core. For example, the first interface circuit provides reading, writing, activating, pre-charging and refreshing operations to the memory cells. A second integrated circuit die, electrically coupled to the first integrated circuit die, comprises a second interface circuit for accessing the memory core via the first interface circuit and for interfacing the memory core to an external circuit. For example, the second interface circuit may comprise a synchronous interface to an external bus. As such, the memory device has two separate die: one for the memory core and a second as an external interface.

In one embodiment, the memory core includes a plurality of memory banks for partitioning the memory cells. A multiplexer, coupled to the memory banks, selects data from one or more of the memory banks. The multiplexer is located generally near an edge of the first integrated circuit die. The first integrated circuit die further comprises data input/output ("I/O") pads and a plurality of bond wires that couple the multiplexer to the I/O pads. The I/O pads are located essentially adjacent to the multiplexer near an edge of the first integrated circuit die so as to minimize distance of the bond wires.

In another embodiment, a distributed-bank architecture is used to configure the memory device. For this embodiment, the memory cells are partitioned into memory banks. The memory banks comprise a plurality of sub-arrays across the physical sections of the memory core such that a physical section of the memory cells comprises a plurality of sub-arrays associated with different memory banks. A multiplexer selects a memory bank from a physical section.

In some embodiment, the first and second integrated circuit dies are housed in separate packages. In other embodiments, the first and second integrated circuit dies are housed in the same package.

## BRIEF DESCRIPTION OF THE DRAWINGS

**FIG 1** is a block diagram illustrating a conventional DRAM chip.

**FIG. 2** illustrates a typical organization of a 4-bank modern SDRAM.

**FIG. 3** is a block diagram illustrating one embodiment of banks arranged in sub-arrays.

**FIG. 4** illustrates a block diagram of an interface and multiple banks in a DRAM.

**FIG. 5** is a block diagram illustrating a DRAM chip with an interface removed.

**FIG. 6** illustrates one embodiment for a center bonded DRAM core chip.

**FIG. 7** illustrates one embodiment for an edge bonded DRAM core chip.

**FIG. 8** illustrates one embodiment for a concentrated-bank architecture.

**FIG. 9** illustrates one embodiment for a distributed-bank architecture.

**FIG. 10** illustrates one embodiment for a quadrant in a distributed-bank architecture DRAM core chip.

**FIG. 11** is block diagram illustrating one embodiment of a distributed-bank architecture universal DRAM core chip.

**FIG 12** is a block diagram illustrating a distributed-bank architecture DRAM core chip configured to support DDR2 speeds.

**FIG. 13** is a block diagram illustrating a distributed-bank architecture DRAM core chip configured to support DDR speeds and external bus widths from 17 to 32 bits.

**FIG. 14** is a block diagram illustrating a distributed-bank architecture DRAM core chip configured to support DDR speeds and external bus widths from 9 to 16 bits.

**FIG. 15** is a block diagram illustrating a distributed-bank architecture DRAM core chip configured to support DDR speeds and external bus widths from 1 to 8 bits.

**FIG. 16** illustrates one embodiment for a portion of a DRAM core chip that includes a decoder for selecting a mode of operation.

**FIG 17a** is a block diagram illustrating the relationship between the internal data bus rate and the external data bus rate for 4n pre-fetching.

**FIG. 17b** is a block diagram illustrating the relationship between the internal data bus rate and the external data bus rate for burst mode with a length of 4n.

**FIG. 18** is a block diagram illustrating an example multi-chip memory implementation.

**FIG. 19** illustrates techniques for stacking two DRAM core chips behind a single interface chip.

## DETAILED DESCRIPTION

The disclosure of U.S. Provisional Patent Application Ser. No. 60/693,631, entitled "Methods and Apparatus for Integrating Multi-Chip Memory Devices", filed on June 24, 2005, is hereby expressly incorporated herein by reference.

By examining several different DRAM architectures, it is clear that the internal organizations of the DRAMs are quite similar. For example, a x16 (x16 denotes the external data width) 256Mb SDRAM, x16 256Mb DDR SDRAM, and x16 256Mb DDR2 SDRAM consist of:

Control logic block;

Address input register and decoder;

Memory arrays that store the data;

Data selection circuit (I/O gating);

Data read circuit; and

Data write circuit.

It is obvious that most of the blocks are common across all the three architectures. The main differences are in the control logic block (which implements the protocol, among other functions), in the width of the data that is accessed per column address, and in the data I/O section. These are usually considered part of the interface section of the DRAM while the rest of the circuits (address decoder, memory arrays, and data selection) are considered part of the memory core. The core timing parameters are typically specified in absolute units of time (seconds) rather than in terms of clock periods. For example, the Micron 256Mb DDR2 SDRAM data sheet lists the following core timing parameters:

**TABLE 1**

| Speed Grade | $t_{RCD}$ (ns) | $t_{RP}$ (ns) | $t_{RC}$ (ns) |
|---|---|---|---|
| -5E | 15 | 15 | 55 |
| -37E | 15 | 15 | 55 |
| -3 | 15 | 15 | 55 |
| -3E | 12 | 12 | 54 |

Similarly, the Micron 256Mb DDR SDRAM data sheet identifies the following timing specifications:

**TABLE 2**

| Speed Grade | $t_{RCD}$ (ns) | $t_{RP}$ (ns) | $t_{RC}$ (ns) |
|---|---|---|---|
| -75Z | 20 | 20 | 65 |
| -75 | 20 | 20 | 65 |

The Micron 256Mb SDRAM data sheet discloses the following specifications:

**TABLE 3**

| Speed Grade | $t_{RCD}$ (ns) | $t_{RP}$ (ns) | $t_{RC}$ (ns) |
|---|---|---|---|
| -7E | 15 | 15 | 60 |
| -75 | 20 | 20 | 66 |

So, even though the protocol and speed of SDRAM, DDR SDRAM, and DDR2 SDRAM are quite different, it is clear that the internal core or array of all these types of DRAMs has similar characteristics. In fact, we can go even further and observe that all synchronous DRAMs are composed of an asynchronous core and an interface that defines the protocol, synchronous operation, speed, and signaling. The memory core typically comprises ~ 90% - 95% of the total die area.

Current practice is to integrate the memory core and the interface onto a common die. The drawback with this approach is that a change in the protocol, speed, or signaling for example requires a re-design of the entire chip. This is usually very expensive and time consuming, and hence the inability to bring specialty or custom DRAMs to the market quickly and cost-effectively. One embodiment of the invention comprises a multi-chip implementation, wherein one or more DRAM core chips are attached to an interface chip. The interface chip sits between the host electronic system and the DRAM core chips. In other words, the interface chip can be thought of as a "wrapper" that surrounds the DRAM core chips. The partitioning of the conventional DRAM into DRAM core chip and interface chip should preferably be done is such a way that that the functions and circuits that are relatively constant across many different architectures are retained in the DRAM core chip while the functions and circuits that vary between the different architectures are moved to the interface chip.

The DRAM core chip can be designed to be suitable for a large number of markets (i.e. a "universal core"). The interface chip can now be designed to meet the exact needs of a market, and even the exact needs of individual customers in that market. To illustrate, the proposed solution enables the design of an interface chip to meet the exact needs of Nokia for the cell phone market and another interface chip to meet the exact needs of Motorola for the cell phone market.

In order to accommodate the needs of the different markets, the DRAM core chip must be capable of operating across a wide range of frequencies, be capable of supporting high data rates, and must be low cost. In one embodiment, the DRAM core chip is asynchronous, wide, and operates at its natural speed. For the case of modern DRAM cores, the natural speed is between 5ns to 10ns per column access, which is equivalent to 100MHz to 200MHz synchronous operation. That is, a modern DRAM core can keep up with an external memory bus or interface that runs at a speed from 100MHz to 200MHz. So, for the case of a synchronous DRAM that operates at 100MHz to 200MHz and is n-bits wide ($1 \leq n \leq 32$ typically), n bits can be fetched from the DRAM core once every clock cycle. In fact, this is how SDRAMs operate.

Newer synchronous DRAMs run at higher clock speeds. JEDEC defines the DDR SDRAM specification with external data rates of 200MHz, 266MHz, 333MHz, and 400MHz. An even newer specification called DDR2 SDRAM has been defined with external data rates of 400MHz, 533MHz, 667MHz, and 800MHz. Effort is currently underway in JEDEC to define a DDR3 SDRAM specification that spans data rates from 800MHz to 1600MHz. GDDR, GDDR2, and GDDR3 SDRAMs typically run faster than the DDR, DDR2, and DDR3 SDRAMs. However, even though the external data rate has been increasing quite rapidly, the speed of the DRAM core has not kept pace. In order to bridge the gap between the external data rate and the internal core speed, the DRAM industry has adopted a technique called "pre-fetching."

Pre-fetching involves accessing more bits than the external data bus width on every column access. To illustrate, an n-bit wide DDR SDRAM accesses 2n bits every column access. This allows the external data bus to run at 200MHz to 400MHz while the internal memory core runs at 100MHz to 200MHz respectively. **FIG. 1** is a block diagram illustrating a conventional DRAM chip. A DRAM chip 100 comprises a DRAM core 110, Internal Data Bus 120, DRAM interface 130 and External data bus 140. **TABLE 4** shows the concept of pre-fetching for a DRAM chip.

**TABLE 4**

| Protocol | External Data Bus 140 Width | Internal Data Bus 120 Width | External Data Rate (MHz) | Internal Data Rate (MHz) |
|---|---|---|---|---|
| SDRAM | N | n | 66 – 133 | 66 – 133 |
| DDR SDRAM | N | 2n | 200 – 400 | 100 – 200 |
| DDR2 SDRAM | n | 4n | 400 – 800 | 100 – 200 |
| DDR3 SDRAM (proposed) | n | 8n | 800 - 1600 | 100 - 200 |

This implies that the universal DRAM core chip must be sufficiently wide enough to support the data rates required by many different markets. Obviously there is a limit to how wide the universal DRAM core chip can be before it starts to negatively impact the cost of the chip. In general, if the width of the DRAM core chip is so large so as to make either the core chip or the interface chip pad limited (especially the core chip), the cost of this solution would be very high.

Modern DRAMs also feature multiple banks. A bank is a section of the DRAM core that can be accessed independently. The DRAM core is broken up into banks that can be active simultaneously. Within each bank, only one row can be open at any given time. Most DRAMs up to 512Mb densities are organized into 4 banks. 1Gb (and possibly, up to 4Gb) DRAMs are organized into 8 banks but only 4 banks can be activated within a specific time window. This is dictated by power and thermal considerations. So, the universal DRAM core chip must be capable of supporting multiple banks.

Let us consider the internal organization of a x16 256Mb SDRAM. A x16 256Mb SDRAM may have 4 banks, each of which is 64Mb. Each bank can be conceptualized as consisting of 16 sub-arrays, each sub-array being a 8192 x 512 matrix of memory cells. That is, each sub-array has 8192 or 8k rows and 512 columns. So, when a bank is accessed, a particular row is accessed (activated) in each of the 16 sub-arrays in the bank. The row is determined by the row address. After the 16 rows are activated, a particular bit in each row is selected. The bit is specified by the column address. So, on each access to a bank, 16 bits are accessed.

**FIG. 2** illustrates a typical organization of a 4-bank modern SDRAM. The memory cells are arranged into four banks: bank 0 (220), bank 1 (210), bank 2 (230) and bank 3 (240). Each bank contains P x Q x 16 cells (*e.g.*, P = 8192 and Q = 512 for a 256 Mb SDRAM). Each bank has associated word line drivers (275, 280, 285 and 290) and sense amplifiers (255, 260, 265 and 270). The banks are selected through use of MUX 250.

In one embodiment, the banks are organized in sub-arrays. **FIG. 3** is a block diagram illustrating one embodiment of banks arranged in sub-arrays. For this embodiment, each bank has 16 sub-arrays (each sub-array being 8K x 512) because the DRAM is organized as a x16 memory.

Consider the internal organization of a x16 256Mb DDR SDRAM. The x16 256Mb DDR SDRAM is organized similar to the x16 256Mb SDRAM with some changes to the memory core. The more important changes to the core organization are:

Each bank has 32 sub-arrays; and

Each sub-array is now 8192 x 256 matrix (i.e. P = 8192, Q = 256). The reason for having 32 sub-arrays is that DDR SDRAM memory uses a pre-fetching of 2n. Since this is a x16 DDR memory, 32 bits must be accessed from each bank for a read or write operation.

Note that pre-fetching can be done in a number of ways. Consider a memory array that is organized as a P x Q matrix that needs to support 2n pre-fetching. One approach is to divide the P x Q array into two arrays (i.e. two P x Q/2 arrays) and access both arrays in parallel, so that we get 2 bits per column address. Another approach is to not split the array but modify the column decoder so that 2 bits are selected for each column address (in other words, the least significant bit of the column address is not used). Some embodiments of the invention are described that use the first approach. However, the teachings of the present invention are applicable to different pre-fetching implementations.

Looking at the organization of a x16 256Mb DDR2 SDRAM, the x16 256Mb DDR2 SDRAM is organized similar to the x16 256Mb SDRAM (and the x16 256Mb DDR SDRAM). The following identify some of the changes to the memory core:

Each bank has 64 sub-arrays. Each sub-array is now a 8192 x 128 matrix (i.e. P = 8192, Q = 128). The reason for the 64 sub-arrays per bank is that DDR2 SDRAM uses a 4n pre-fetching. Since this is a x16 DDR2 memory (n = 16), 64 bits must be accessed from each bank for a read or write operation.

In all cases (SDRAM, DDR, DDR2), data bits to/from each bank are brought to a multiplexer / de-multiplexer (hereafter referred to as a MUX), which in turn is connected to the external DQ pins. This MUX is typically in the middle of the DRAM chip. **FIG. 4** illustrates a block diagram of an interface and multiple banks in a DRAM. As shown in **FIG. 4**, bank 0 (410), bank 1 (420), bank 2 (430) and bank 3 (440) are accessed by interface 450. Note that for SDRAM, m = n, for DDR SDRAM, m = 2n, DDR2 SDRAM, m = 4n, and DDR3 SDRAM (proposed), m = 8n, Also note that the data MUX is typically part of the interface.

This arrangement works well for a conventional DRAM since everything is on a single die. However, in one embodiment of the invention, the interface is on a separate die. If we were to just move the interface alone (to another die) without disturbing the memory core, then the number of I/O pads on both the memory core chip and the interface chip will become quite large, as shown in **TABLE 5** below for a x16 4-bank implementation. **FIG. 5** is a block diagram illustrating a DRAM chip with an interface removed. For this embodiment, bank 0 (510), bank 1 (520), bank 2 (530) and bank 3 (540) are coupled to I/O pads 550, 560, 570 and 580, respectively. If we look at only the data pins and ignored the address, command, power and ground pins, we can see that the number of data signals that have to go off-chip between the DRAM core chip and the interface chip is 4m.

**TABLE 5**

| Protocol | External Data Bus Width (n) | m | Total Number of Off-Chip Data Pins (4m) |
|---|---|---|---|
| SDRAM | 16 | 16 | 64 |

| | | | |
|---|---|---|---|
| DDR SDRAM | 16 | 32 | 128 |
| DDR2 SDRAM | 16 | 64 | 256 |
| DDR3 SDRAM | 16 | 128 | 512 |

So, it is quite obvious that removing the interface without disturbing the rest of the memory core quickly leads to a very large number of off-chip connections, especially for wider external data bus widths and higher data speeds (because, the amount of pre-fetching will increase with higher data speeds). Under these conditions, either the DRAM core chip or the interface chip or both will become pad limited, which will increase the cost of the total solution.

In one embodiment, in order to reduce the number of off-chip connections between the DRAM core chip and the interface chip, some part or all of the multiplexing of the data I/O from the banks is done in the core chip itself.

One option is to route all the data bits from each bank to a central MUX, and then connect the other side of the MUX to off-chip drivers. This is quite similar to the current practice for center bonded DRAMs. **FIG. 6** illustrates one embodiment for a center bonded DRAM core chip. For this example, integrated circuit 600 includes bank 0 (610), bank 1 (620), bank 2 (630) and bank 3 (640) coupled to MUX 650. MUX 650 is connected to substrate bonding pads 670 via bond wires 680 through I/O pads 660.

The drawback with this approach is that bond wires 680 that connect I/O pads 660 on the DRAM core die to the substrate bonding pads 670 become quite long. Long bond wires have significant inductance and limit the speed at which the memory chip can operate.

In another embodiment, edge bonding for the core chip is used. **FIG. 7** illustrates one embodiment for an edge bonded DRAM core chip. For this embodiment, integrated circuit 700 includes bank 0 (710), bank 1 (720), bank 2 (730) and bank 3 (740). I/O pads 750, located on the silicon die 705, are connected to the substrate bonding pads 760 via bond wires 780.

If the DRAM core was organized with one bank per quadrant, then the number of data I/O pads will be equal to 4m as illustrated previously. The other option is to route the data bits from each bank to a centrally located MUX (as shown in **FIG. 6**) and then route the signals from the other side of the MUX to the periphery of the die. However, this means that the data signals will have to traverse the die twice – once from the bank to the central MUX and once from the central MUX to the periphery. This increases routing complexity, may possibly require an extra metal layer on the die (higher cost), and adds to the latency of the memory core.

In another embodiment, an inventive "Distributed-Bank" architecture is used. In this architecture, a bank is distributed (or spread) across all 4 quadrants instead of concentrating a bank in only one quadrant. Using this architecture, data MUXs, located in all 4 quadrants, select the appropriate bank, and the data signals corresponding to the selected bank can be easily routed to the periphery of the chip.

**FIG. 8** illustrates one embodiment for a concentrated-bank architecture. For the purpose of this illustration, a x16, 4-bank, 256Mb DDR2 SDRAM core is used. However, any type of DRAM, with different external data widths, different number of banks, different density, and different amount of pre-fetching may be used without deviating from the spirit or scope of the invention. As previously shown, each bank (810, 820, 830 and 840) in a x16, 4-bank, 256Mb DDR2 SDRAM consists of 64 sub-arrays, with each sub-array organized as a 8192 x 128 array of memory cells.

**FIG. 9** illustrates one embodiment for a distributed-bank architecture. For this embodiment, the DRAM core chip is divided into four quadrants (910, 920, 930 and 940). Each quadrant includes a portion of a bank.

**FIG. 10** illustrates one embodiment for a quadrant in a distributed-bank architecture DRAM core chip. As discussed previously, there are 64 sub-arrays per quadrant with each sub-array being a 8192 x 128 matrix. Instead of assigning all 64 sub-arrays in a single quadrant to a single bank in the concentrated-bank architecture, there are16 sub-arrays to each of the 4 banks within a single quadrant in the distributed-bank architecture. In the distributed-bank architecture embodiment, local data MUXs are located in each quadrant to select one of the four banks.

**FIG. 11** is block diagram illustrating one embodiment of a distributed-bank architecture universal DRAM core chip. For this embodiment, the banks of DRAM cells are distributed among quadrant 1110, 1120, 1130 and 1140. An address decoder 1150, located in the center of the chip, controls word line drivers 1155, 1164, 1170 and 1176 in quadrants 1110, 1120, 1130 and 1140 respectively. The data from the bank sub-arrays are output to sense amplifiers (1157, 1166, 1172 and 1178) and input to the respective bank select MUXs (1160, 1168, 1174 and 1180). The data is then routed to data I/O pads 1162 located in the proximity for each of the quadrants.

Since 64 data bits are accessed from the core chip for every read or write operation, the interface chip may be designed or configured to act similar to a x16 DDR2 SDRAM, x8 DDR2 SDRAM, x4 DDR2 SDRAM, x2 DDR2 SDRAM, or x1 DDR2 SDRAM. In fact, the interface chip may be designed to support any data width between x1 and x16 when operating in a 4n pre-fetch mode.

The distributed-bank architecture is flexible enough to support protocols like SDRAM, DDR SDRAM, DDR2 SDRAM, and DDR3 SDRAM. For example, the DRAM core chip shown in **FIG. 11** may be configured or used as shown in **FIG. 12** to support DDR2 speeds. **FIG 12** is a block diagram illustrating a distributed-bank architecture DRAM core chip configured to support DDR2 speeds. MUX 1210 selects 64 bits of data for one of the banks (1220, 1230, 1240 and 1250). MUX 1210 represents the data MUXs located in all four quadrants on the DRAM core chip.

The DRAM core chip shown in **FIG. 11** may also be configured or used as shown in **FIG 13** to support DDR SDRAM speeds when it is operated in a 2n pre-fetch mode. **FIG. 13** is a block diagram illustrating a distributed-bank architecture DRAM core chip configured to support DDR speeds and external bus widths from 17 to 32 bits. MUX 1310 selects 64 bits of data for one of the banks (1320, 1330, 1340 and 1350). The mode of operation shown in FIG. 13 may be used with the appropriate interface chip to support external data widths between (and inclusive) of 17 and 32 in a 2n pre-fetch mode. The same DRAM core chip may be used with the appropriate interface chip to support external data widths between (and inclusive) 9 and 16 in a 2n pre-fetch mode when operated as shown in **FIG. 14**. **FIG. 14** is a block diagram illustrating a distributed-bank architecture DRAM core chip configured to support DDR speeds and external bus widths from 9 to 16 bits. For this embodiment, MUX 1410 selects 32 bits of data for one of the banks (1420, 1430, 1440 and 1450).

Also, the same DRAM core chip can be used with the appropriate interface chip to support external data widths between (and inclusive of) 1 and 8 in a 2n pre-fetch mode. **FIG. 15** is a block diagram illustrating a distributed-bank architecture DRAM core chip configured to support DDR speeds and external bus widths from 1 to 8 bits. For this embodiment, MUX 1510 selects 16 bits of data for one of the banks (1520, 1530, 1540 and 1550).

From these architectures, the internal data bus width (the width of the bus between the DRAM core chip and the interface chip) may be configured to match the amount of pre-fetching required (which is determined by the external data rate) and the width of the external data bus. The external data bus is the bus from the interface chip to the ASIC or memory controller. The DRAM core chip as shown in **FIG. 11** may be configured to support the following modes and requirements shown in **TABLE 6.**

**TABLE 6**

| Internal Data Bus Width | Pre-Fetching Used | Minimum External Data Bus Width | Maximum External Data Bus Width |
|---|---|---|---|
| 64 | 8n | 1 | 8 |
| | 4n | 1 | 16 |
| | 2n | 1 | 32 |
| | 1n | 1 | 64 |
| 32 | 8n | 1 | 4 |
| | 4n | 1 | 8 |
| | 2n | 1 | 16 |
| | 1n | 1 | 32 |
| 16 | 8n | 1 | 2 |
| | 4n | 1 | 4 |
| | 2n | 1 | 8 |
| | 1n | 1 | 16 |

Note that:

The proposed DDR3 SDRAM is an example of an 8n pre-fetch protocol;

DDR2 SDRAM is an example of a 4n pre-fetch protocol;

DDR SDRAM is an example of a 2n pre-fetch protocol; and

SDRAM is an example of a 1n pre-fetch protocol (i.e. no pre-fetching needed).

Again, for the DRAM core chip shown in **FIG. 11**, 3 modes of operation may be defined and a 2-bit binary code may be assigned to represent them as shown in **TABLE 7**.

**TABLE 7**

| Mode of Operation (Mode[1:0]) | Internal Data Bus Width |
|---|---|
| 00 | 64 |
| 01 | 32 |
| 10 | 16 |
| 11 | Undefined/Reserved |

These two bits (Mode[1:0]) may be inputs to the DRAM core chip so that the internal data bus width is selected through external means. For example, the Mode[1:0] inputs to the core chip may be selected by means of fuses on the core chip or on the interface chip, by means of pull-up or pull-down resistors in the package of either chip (or in the common package) or on the printed circuit board, or may be driven by a register on the interface chip, or may be part of the address input to the core chip.

Let us assume that the Mode[1:0] inputs to the DRAM core chip are controlled by a register in the interface chip. **FIG. 16** illustrates one embodiment for a portion of a DRAM core chip that includes a decoder for selecting a mode of operation. The decoder 1610 in the core chip is aware of the Mode[1:0] inputs as shown in **FIG. 16.**

Note that the embodiments disclosed below are based on the DRAM core chip shown in **FIG. 11** only for the purpose of explaining the concept, and that the following embodiments are applicable to DRAM core chips of different densities, number of banks, internal organization, and number of sub-arrays. For the purpose of simplicity, only bank 0 is shown being accessed in the different modes of operation.

The mode decoder truth table is shown below in **TABLE 8**. In **TABLE 8** below,

RA = Row Address

X = Don't Care

H = Asserted

L = Not Asserted

**TABLE 8**

| Mode[1:0] | RA[14] | RA[13] | Bank0a_En | Bank0b_En | Bank0c_En | Bank0d_En |
|---|---|---|---|---|---|---|
| 00 | X | X | H | H | H | H |
| 01 | X | L | H | H | L | L |
| | X | H | L | L | H | H |
| 10 | L | L | H | L | L | L |

| | | | | | | |
|---|---|---|---|---|---|---|
| | L | H | L | H | L | L |
| | H | L | L | L | H | L |
| | H | H | L | L | L | H |

Based on the techniques of the present invention, a universal DRAM core chip, such as the embodiment shown in **FIG. 11**, may be configured to support a wide variety of data speeds and widths. For example, the core chip shown in **FIG. 11** may support data rates up to 8*f MB/s, where f is the maximum clock rate (in MHz) at which the DRAM core can run in sync with the external data bus without using pre-fetching (i.e. using a pre-fetching of 1n). For modern DRAM processes and designs, f is typically between 100MHz and 200MHz. So, the DRAM core chip shown in **FIG. 11** supports maximum data rates between 800MB/s and 1600MB/s (1.6GB/s).

In order to build custom memory cost-effectively, it is imperative that the DRAM core chip be used in a variety of markets. This will reduce the cost of the core chip because of economies of scale. Since the memory core is typically 90% to 95% of the silicon area, the overall cost may be lowered. Here, we can make two observations:

Some markets for the universal DRAM core chip value low power at the expense of performance (e.g. cell phones and other handheld devices) whereas other markets will sacrifice power to achieve higher speed (e.g. PC graphics and game console markets).

Semiconductor fabrication process is inherently statistical in nature. That is, if we fabricate a statistically significant number of identical chips, some of the chips will only be capable of operating below the target speed, some of the chip will be capable of operating at the target speed, and some of the chips will be capable of operating above the target speed. These are known in the industry as slow, typical, and fast parts respectively. Usually, the fast parts are sold at a price premium over the other parts while the slow parts are sold at lower prices compared to the typical parts.

However, slow parts typically consume less power than the typical parts, which in turn typically consume less power than the fast parts. So, if we can sort the DRAM core chips according to their maximum speeds of operation (usually called "speed binning") before they are attached to the interface chips, we can:

Attach interface chips designed for low power markets (e.g. an SDRAM-like interface for the cell phone market) to the slow core parts;

Attach interface chips designed for the high performance/speed markets (e.g. a GDDR3-like interface for the game console market) to the fast core parts; and

Attach interface chips designed for markets sensitive to both power and performance (e.g. a DDR-like interface for the server market) to the typical core parts.

This allows us to maximize the ASP (average selling price or average sales price) of all the solutions since all the core chips have natural homes.

Speed binning of memory chips is typically done after it is packaged. Note that it is possible to do a simple speed sort of the memory chips at the wafer level itself. In order to do a speed sort or speed bin, we need to use ATE (automatic test equipment), also known as tester.

When we speed bin the DRAM core chips, we need to measure the time required for basic operations like Read, Write, Activate (open one or more pages), Precharge (close one or more pages), and Refresh. To satisfy this requirement, the DRAM core chips, defined by the present invention, are fully functional asynchronous DRAM chips capable of stand-alone operation. In other words, the DRAM core chips contain all the necessary circuits and capabilities needed to access the internal array used to store the data.

Memory makers, especially DRAM manufacturers, build redundancy into the memory core. For example, if the memory array is to be organized as P x Q (P rows and Q columns), the actual array is designed as (P + i) x (Q + j), where i and j are small compared to P and Q respectively. This allows the memory makers to replace up to i defective rows in the main array with the redundant rows, and up to j defective columns in the main array with the redundant columns. With the help of the redundant rows and columns, memory makers can increase the yield (i.e. the percentage of fully functional chips) to ≥90%. In a typical DRAM manufacturing flow, the individual dies on a wafer are tested at low speed and the partially functional dies (i.e. those with some defective rows and/or columns) are marked. The defective rows and/or columns on these marked dies are replaced with the redundant rows and/or columns respectively.

However, a die that uses the redundant rows and/or columns (because it had some defective rows and/or columns) will be slower than a die that does not use redundant rows and/or columns. This is due to the nature of how redundancy is built into the memory and how it is enabled. Therefore:

Interface chips that are designed for high performance markets may be attached to the DRAM core dies that do not have any defective rows and/or columns in the main array.

In another embodiment, the defective rows and/or columns of memory core dies are not replaced with the redundant rows and/or columns, but are configured to operate the memory core dies as (P / y) x (Q / z), where y and z are preferably powers of 2 (including $2^0 = 1$). These DRAM core chips may then be attached to interface chips that are designed for high performance markets.

As we have seen previously, the DRAM makers use pre-fetching to support higher external data rates. For example, DDR2 SDRAM uses 4n pre-fetching. This means that for an n-bit wide external data bus, 4n data bits are accessed from the memory core for every read or write. In a conventional DRAM (where the memory core and the interface are on the same die), increasing the amount of pre-fetching increases the amount of metal interconnects on the die, which has a modest impact on the cost. In the invention described herein, increasing the amount of pre-fetching may make either the memory core chip or the interface chip or both pad limited. Being pad limited can increase the cost substantially.

Burst mode is another technique that can be used to increase the data rate of memory chips. In burst mode, the memory chip reads or writes multiple data bits per column address. For example, an n-bit wide (external data bus width) memory chip that is configured for a burst mode of 4n will access 4n bits from the memory core for a given column address. So this is quite similar to a 4n pre-fetch except that in burst mode, the same data wires are used. In other words, in a memory chip that supports 4n burst mode (but not 4n pre-fetching), the internal data bus between the memory core and the interface is only n-bits wide. Each line in the internal bus carries 4 data bits that are separated in time.

The difference between pre-fetching and burst mode is shown in **FIGS. 17a** and **17b**. **FIG 17a** is a block diagram illustrating the relationship between the internal data bus rate and the external data bus rate for 4n pre-fetching. Memory core 1710 is coupled to memory interface 1720 via internal data bus 1715 at 4n @ $f_1$, Hz data rate. The memory interface 1720 is coupled to external data bus 1725, and under these conditions, the external data bus operates at a n @ $4*f_1$, Hz data rate. **FIG. 17b** is a block diagram illustrating the relationship between the internal data bus rate and the external data bus rate for burst mode with a length of 4n. Memory core 1730 is coupled to memory interface 1750 via internal data bus 1740 at n @ 4*$f_2$, Hz data rate. The memory interface 1750 is coupled to external data bus 1760, and for the burst mode of operation, the external data bus operates at a n @ 4*$f_2$, Hz data rate.

Typically, pre-fetching will provide higher external data rates than burst mode. However, burst mode does not increase the amount of off-chip connections between the core chip and the interface chip. So, in some embodiments, it is preferable to design the DRAM core chip of this invention with burst mode capability.

As mentioned previously, one of the aspects of this invention is the ability to test and speed bin the memory core chips and then attach the appropriate interface chips. Testing and speed binning of the DRAM core chip is usually done on a tester. This requires the core chip to have sufficiently strong output drivers to drive the inputs of the tester, which are usually some distance (several inches) from the outputs of the core chip. However, in the normal mode of operation, the inputs of the interface chip will be much closer (< 1") to the outputs of the core chip. So, it is not necessary to have strong output drivers in the core chip in the normal mode of operation. In order to satisfy both requirements, in some embodiments the DRAM core chip preferably has output drivers whose strength or drive capability is adjustable. For example, the core chip may have, by default, normal strength output drivers that are capable of driving signals across some distance to the inputs of the tester. However, when an interface chip is attached to the core chip, a signal from the interface chip decreases the drive strength of the core chip's output drivers.

In some embodiments, the output drivers of interface chip that interface with the DRAM core chip have similar adjustable drive strength capability. This allows testing of the interface chips separately prior to attaching them to the core chips. Note that the adjustable drive strength drivers are not necessarily required on the interface chip on the pins that interface to the electronic host system. However, it is preferable to have the adjustable strength capability on these drivers as well so that the drive strength can be tailored to the requirements of the system or external world. As with the DRAM core chips, the strength of the output drivers on the interface chip that communicate with the core chip are preferably controlled by a signal from the core chip.

**FIG. 18** is a block diagram illustrating an example multi-chip memory implementation. The solution includes a DRAM core chip 1810 and an interface chip

1820. In some embodiments, the main characteristics of the DRAM core chip of this invention are:

Asynchronous or synchronous DRAM that is capable of stand-alone operation;

A plurality of DRAM core chips may be coupled together with one or more interface chips;

A plurality of interface chips may be coupled together with one or more DRAM core chips;

The interface on the DRAM core chip may include a custom and/or industry standard interface;

Has address inputs (bank address, row address, column address – row and column address can be on separate inputs or multiplexed on same pins);

Has command inputs like address strobes, read/write, output enable, and data masks);

Has control inputs that determine mode of operation – examples are inputs that determine the width of the internal data bus (bus between the memory core chip and interface chip) and inputs that determine the strength of the output drivers;

Has control outputs that determine some aspect of the functions performed by the interface chip;

Internal data bus width $\geq$ external data bus width (bus from interface chip to memory controller or ASIC);

Optional burst mode capability;

Adjustable drive strength on output drivers;

Capable of well-defined standard operations like Read, Write, Activate, Precharge, and Refresh that can be clearly characterized in terms of speed; and

May be tested, burnt in, and speed binned independently (i.e. in stand-alone mode).

In some embodiments, the main characteristics of the interface chip of this invention are:

Implements the protocol used by the memory controller;

Interface chip implements an industry standard protocol like DDR SDRAM, DDR2 SDRAM, GDDR2 SDRAM, etc.;

Interface chip implements an industry standard protocol with custom extensions (e.g. GDDR2 SDRAM with extensions as specified by mutual agreement with one or more customers);

Interface chip implements a fully custom protocol as specified by one or more customers or a fully custom protocol developed in-house;

Interface chip operates as a transformer to convert protocols from the external interface to the interface of the DRAM core chip (*e.g.*, synchronous to asynchronous and asynchronous to synchronous);

Interface chip determines the signaling used by the external interface;

For example, single ended, pseudo-differential, fully differential;

For example, push-pull outputs, open drain/collector outputs;

For example, asynchronous, synchronous, source synchronous, SerDes-like where clock is encoded/embedded in the data stream;

Determines the width of the external data bus;

Determines the speed of operation of the memory chip (by memory chip, we mean the combination of the DRAM core chip and the interface chip);

Determines the pin out of the memory chip as seen by the external world;

Allows the pin out of the memory chip to better match the pin out of the ASIC / memory controller to reduce the board routing complexity;

Implements special or custom functions and modes of operation;

Example, special power management functions and operating modes; and

Example, special error detection and error correction capabilities, as well as other types of redundancy capabilities and functions.

The DRAM core chip and the interface chip of this invention may be attached together in a number of different ways:

One or more DRAM core chip dies and one or more interface chip dies may be electrically connected to each other and the whole combination be put into a single package (e.g., a single DRAM core chip die with a single interface chip die, multiple DRAM core chip dies with a single interface chip die, or a single DRAM core chip die with multiple interface chip dies).

The core chip die may be put in a separate package, and then the interface chip die may then be electrically attached to the package containing the core chip die;

The interface chip die may be put in a separate package, and then the core chip die may then be electrically attached to the package containing the interface chip die;

The core chip die can be put in a separate package; the interface chip die may be put in a separate package; and the two packages can be electrically attached to each other;

Multiple DRAM core chip dies may be put in a separate package; the interface chip die may be put in a separate package; and the two packages can be electrically attached to each other;

A DRAM core chip die may be put in a separate package; multiple interface chip dies may be put in a separate package; and the two packages can be electrically attached to each other;

The DRAM core chip die and the interface chip die may be electrically attached in any way without deviating from the spirit or scope of the invention.

One aspect of this invention is that a multi-chip DRAM that is built according to this invention, might have higher cost than a traditional DRAM, especially if the DRAM core chip die and the interface chip die were packaged separately and then attached to each other. This is due to the cost associated with the extra package. One way to ameliorate this is to put multiple DRAM core chip dies in a single package. For the purpose of this discussion, we shall consider putting two DRAM core chip dies in a single package, each die being a 256Mb density device. A typical DRAM manufacturing process might have the following sequences after the wafer has been fully processed:

DRAM dies on a wafer are tested at low speed and dies with defective rows and/or columns are marked;

The defective rows and/or columns are replaced with redundant rows and/or columns;

Wafer is diced into individual dies, which are then packaged;

Packaged parts are tested for functionality – parts damaged by the packaging process are eliminated;

Tested packaged parts undergo long term burn in to eliminate infant mortality parts; and

Burnt in parts are optionally tested again for functionality and shipped.

So, if we place two 256Mb DRAM core dies in a single package, the following 3 bins may be generated after the parts have been packaged and burnt-in:

Bin A – both the DRAM core chip dies are functional, so the total capacity is 512Mb

Bin B – only one of the DRAM core chip dies is functional, so the total capacity is 256Mb

Bin C – neither of the DRAM core chip dies is functional, so the total capacity is 0Mb

The bin C parts should be discarded. We can now use the bin B parts for those markets and/or customers who require only 256Mb devices. For example, a handheld device manufacturer might require only a 256Mb DRAM. So, the bin B parts can be attached to the interface chips designed for this manufacturer. Other markets and/or manufacturers might require 512Mb devices. For example, a network router manufacturer might need 512Mb DRAMs. So, we can use bin A parts for this manufacturer by attaching the appropriate interface chips to the bin A parts. This concept can be extended to cover more than two DRAM core chip dies in a single package as well as DRAM core chip dies of all densities.

As disclosed above, some embodiments cover the idea of attaching multiple DRAM core chips to a single interface chip. The core chips may be attached to the interface chip in a number of different ways. Some of these ways are described below and in **FIG. 19**. Again, for the purpose of illustration, we shall assume that two 4-bank 256Mb DRAM core chips (**FIG. 11**) are attached to the interface chip. Each of the core chips has a 64-bit wide data bus designed to connect to the interface chip. Note that the idea explained below can be applied to DRAM core chips with different number of banks, density, data bus width, etc.

In one method (1910), the row addresses are used to select the DRAM core chip. In a second method (1920), bank addresses are used to select DRAM core chips. In a third method (1930), the column addresses are used to select the DRAM core chip.

The two DRAM core chips are attached to the interface chip such that only one of the core chips is accessible at any given time. That is, the two core chips look to the electronic host system as a single 512Mb DRAM with 4 banks. This implies that the interface chip will use the row address to select one or the other core chip.

The two DRAM core chips are attached to the interface such that both of the core chips are accessible at any given time, and that the two chips look to the external world electronic host system as a single 512Mb DRAM with 8 banks. This implies that the interface chip will use the bank address to select one or the other core chip.

The two DRAM core chips are attached to the interface chip such that both of the core chips are accessible at any given time, and that the two chips look to the electronic host system as a single 512Mb DRAM with 4 banks. The interface chip uses the column address to select one or the other core chip. Note that in other embodiments an interface chip always accesses both the core chips in parallel, so that the data bus between the core chips and the interface chip becomes 128-bits wide. For this embodiment, the interface chip doubles the external data rate. In other words, the amount of pre-fetching has been doubled.

In other embodiments the multi-chip solution is configured such that the attachment of the DRAM core chips to the interface chip is programmable. For example, the customer may choose between one of the three methods listed above to attach two 4-bank, 64-bit wide, 256Mb DRAM core chips to an interface chip by programmable means. These means include using fuses on the interface chip or the core chips, pull-up or pull-down resistors on the package substrates or printed circuit board, or by means of a register on the interface chip or core chips. However, any way of attaching the DRAM core chips to the interface chip may be accomplished without deviating from the spirit or scope of the invention.

Other embodiments of the invention include building redundant memory systems by attaching multiple DRAM core chips to an interface chip. For example, when more than one core chip is attached to an interface chip, redundancy is added by several means including:

Storing identical copies of the data in corresponding locations of each core chip when data is written to the memory by the memory controller; when the data is read back by the memory controller, the interface chip can read the multiple copies of the data from the different core chips and select the correct copy and transmit it to the memory controller. The correct copy can be determined by means like majority voting, and/or by the use of parity or ECC bits;

Using (n + m) bits to store n data bits.

Another aspect of this invention is placing memory core chips of different types behind a common interface chip. For example, we can place any combination of DRAM core, SRAM (Static Random Access Memory), and Flash chips behind a common interface chip. Multi-Chip Package (MCP) memory solutions are fairly common in the cell phone and handheld markets today. The issue with current MCP solutions is that each of these memories (DRAM, SRAM, Flash) has different interfaces, which complicates the design of the memory controller, the packaging, and the board routing. Placing any possible combination of DRAM core chip, SRAM, and Flash behind a common interface chip simplifies the memory controller design since the idiosyncrasies of each of these memory types is hidden from the memory controller. In addition, the board routing is simplified.

Another aspect of this invention is placing a large and slow memory as well as a smaller and faster memory behind a common interface chip and using the faster memory as a cache for the slower memory. For example, an SRAM chip might be used as the cache for a DRAM core chip or a DRAM core chip can be used as the cache for a Flash chip. The cache management logic may be built into the interface chip so that the cache is transparent to the memory controller. Alternately, the cache may be made visible to the memory controller and managed by the memory controller. Let us consider the case of an interface chip that has been designed to interface to one or more DRAM core chips and an SRAM chip. The SRAM chip can be used to cache the rows in the DRAM core chips that were recently opened. The SRAM may be used to cache the entire contents of the recently opened rows in the DRAM or cache part of the contents of the recently opened rows in the DRAM. The properties of the cache (associatively of the cache lines, mapping between DRAM rows and SRAM cache lines, etc.) may be determined by programming certain registers in the interface chip. By storing data that has a high likelihood of being accessed in the near future, system performance is improved.

Using a cache also allows the interface chip to do speculative pre-fetching of data from the DRAM core chip (and storing it in the SRAM cache chip), which again improves system performance. In addition, the interface chip may operate with SRAM chips (that are used as caches) of different densities. This allows the same interface chip to be used across several different segments within a market. For example, a DRAM / SRAM combination memory device may includes a common interface where the DRAM capacity is 512Mb and the SRAM capacity ranges from 0 to 32Mb. This allows a DRAM supplier to ship the 512Mb DRAM + 32Mb SRAM combination in the high performance segment of the market, ship a 512Mb DRAM + 8Mb SRAM combination in the mainstream segment of the market, and ship a 512Mb DRAM (no SRAM cache) device in the value segment of the market.

The techniques of the present invention are not just applicable to DRAM. As such, the DRAM core chip need not necessarily be a trench-capacitor or stacked-capacitor device. The present invention is applicable to a variety of memory technologies like MRAM (Magnetic RAM), FRAM (Ferro-electric RAM), Ovonics memory, molecular memory (e.g. memory technology developed by ZettaCore), carbon nanotube memory (e.g. memory technology developed by Nantero Inc.), etc.

Another aspect of this invention is that it can be used with DRAM core chips that have different architectures like FCRAM (Fast Cycle RAM), RLDRAM (Reduced Latency DRAM), ESDRAM (Enhanced SDRAM).

Some embodiments of invention allow the use of a common memory core across a wide range of markets while varying the interface of the memory core according to the market and customer needs. It also allows the interface to be kept constant while changing the memory core behind the interface to address the needs of the different segments within a market.

Although the present invention has been described in terms of specific exemplary embodiments, it will be appreciated that various modifications and alterations might be made by those skilled in the art without departing from the spirit and scope of the invention.

## CLAIMS

What is claimed is:

1. A memory device comprising:

at least one first integrated circuit die comprising:

memory core comprising a plurality of memory cells;

first interface circuit for accessing said memory cells of said memory core; and

at least one second integrated circuit die, electrically coupled to said first integrated circuit die, comprising a second interface for accessing said memory core via said first interface circuit and for interfacing said memory core to an external circuit.

2. The memory device as set forth in claim 1, further comprising a plurality of first integrated circuit dies.

3. The memory device as set forth in claim 1, further comprising a plurality of second integrated circuit dies.

4. The memory device as set forth in claim 1, further comprising:

first package for housing said at least one first integrated circuit die; and

second package for housing said at least one second integrated circuit die.

5. The memory device as set forth in claim 1, further comprising a single package for housing said first and second integrated circuit dies.

6. The memory device as set forth in claim 1, further comprising a single package for housing said at least one first integrated circuit die and for housing a plurality of second integrated circuit dies.

7. The memory device as set forth in claim 1, further comprising a single package for housing a plurality of first integrated circuit dies and for housing said at least one second integrated circuit die,

8. The memory device as set forth in claim 1, further comprising:
first package for housing a plurality of first integrated circuit dies; and
second package for housing said at least one second integrated circuit die.

9. The memory device as set forth in claim 1, wherein said second interface of said second integrated circuit die for converting protocols between said external circuit and first interface of said first integrated circuit.

10. The memory device as set forth in claim 9, wherein said protocols comprise different protocols.

11. The memory device as set forth in claim 9, wherein said second interface of said second integrated circuit die for converting between synchronous and asynchronous protocols.

12. The memory device as set forth in claim 9, wherein said second interface of said second integrated circuit die for converting between custom and industry standard protocols.

13. The memory device as set forth in claim 1, wherein said first interface circuit of said first integrated circuit die provides a read operation to said memory cells.

14. The memory device as set forth in claim 1, wherein said first interface circuit of said first integrated circuit die provides a write operation to said memory cells.

15. The memory device as set forth in claim 1, wherein said first integrated circuit of said first integrated circuit die provides activating, pre-charging and refreshing operations to said memory cells.

16. The memory device as set forth in claim 1, wherein said first integrated circuit die comprises a plurality of memory banks for partitioning said memory cells.

17. The memory device as set forth in claim 16, wherein said memory banks comprise a plurality of sub-arrays arranged in a distributed bank architecture across a plurality of physical sections such that a physical section of said memory cells comprises a plurality of sub-arrays associated with different memory banks.

18. The memory device as set forth in claim 1, wherein said memory cells comprise at least one non-volatile memory cell.

19. The memory device as set forth in claim 1, wherein said memory cells comprise at least one volatile memory cell.

20. A memory device comprising:

first integrated circuit die comprising:

memory core comprising a plurality of memory cells;

first interface circuit coupled to said memory cells, for dynamically configuring a data rate for transferring data between said memory cells and said first interface circuit; and

second integrated circuit die, electrically coupled to said first integrated circuit die, comprising a second interface for accessing data from said memory core via said first interface circuit and for interfacing said memory core to an external circuit.

21. The memory device as set forth in claim 20, wherein:

said memory device comprises an internal data bus for coupling data between said first integrated circuit die and said second integrated circuit die; and

said configurable internal data rate comprises a configurable data width for said internal data bus.

22. The memory device as set forth in claim 20, wherein said configurable internal data rate comprises a configurable amount of data for pre-fetching.

23. The memory device as set forth in claim 20, wherein said first integrated circuit die further comprising at least one input for programming said internal data rate.

24. The memory device as set forth in claim 20, further comprising an external data bus comprising an external data rate for accessing data external to said memory device, wherein said configurable internal data rate permits selecting an internal data rate compatible with said external data rate.

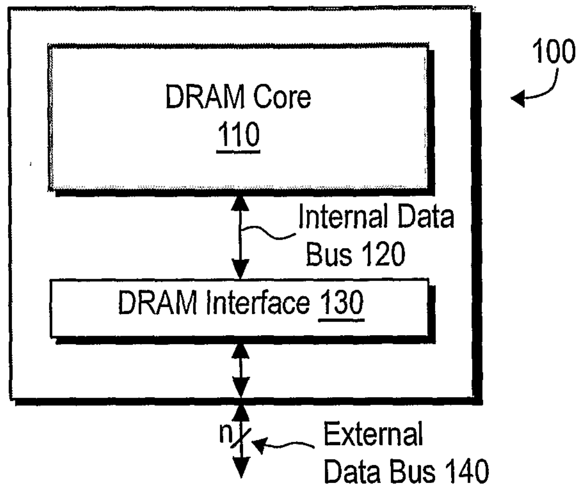

DRAM Core
110
Internal Data
Bus 120
DRAM Interface 130
n
External
Data Bus 140
100

FIG. 1

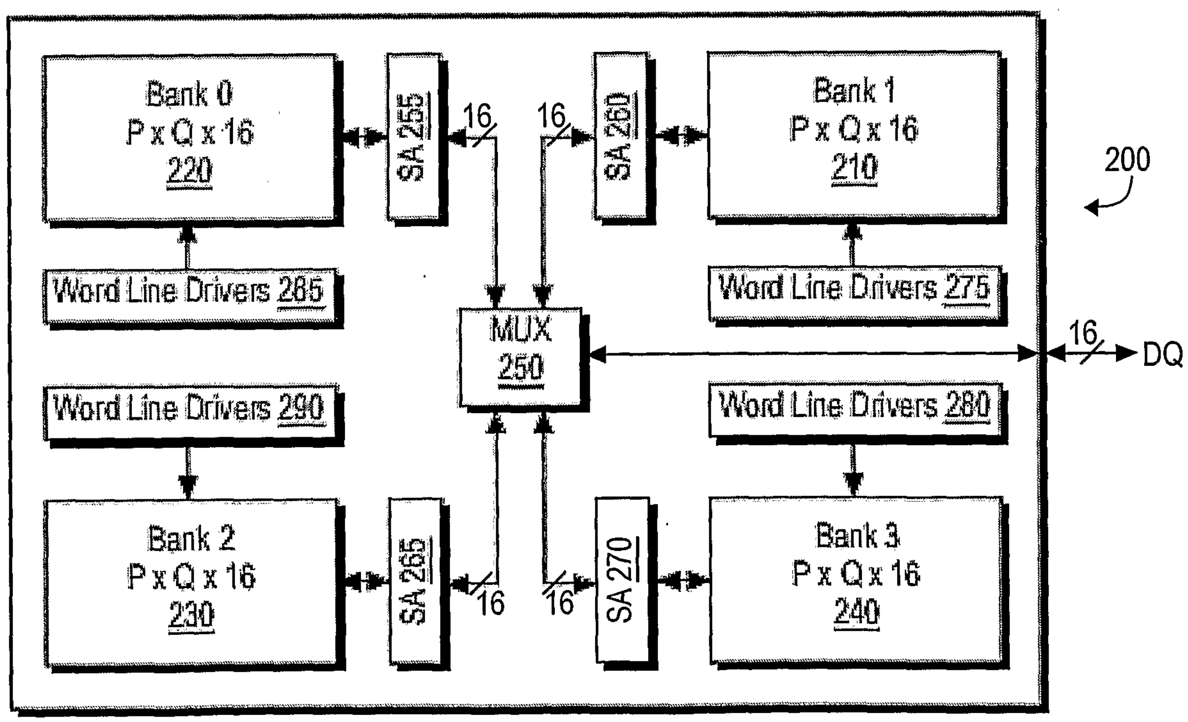

Bank 0
P x Q x 16
220
SA 255
16
16
SA 260
Bank 1
P x Q x 16
210
200
Word Line Drivers 285
Word Line Drivers 275
MUX
250
16
DQ
Word Line Drivers 290
Word Line Drivers 280
Bank 2
P x Q x 16
230
SA 265
16
16
SA 270
Bank 3
P x Q x 16
240

FIG. 2

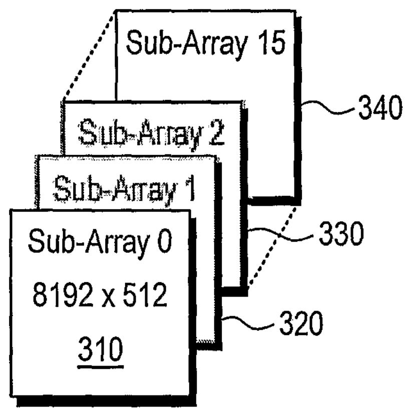

Sub-Array 15
340
Sub-Array 2
Sub-Array 1
330
Sub-Array 0
8192 x 512
310
320

FIG. 3

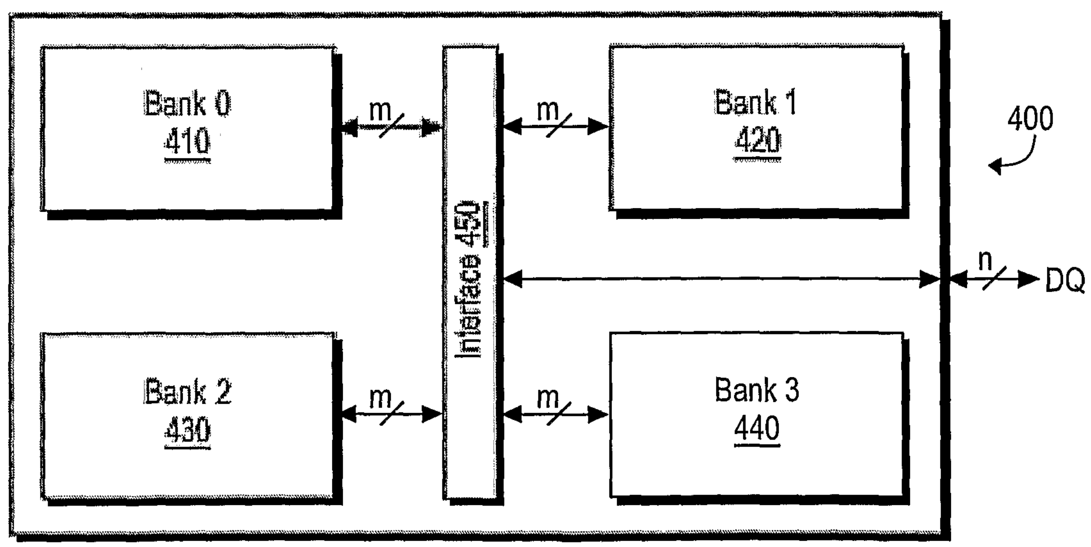

Bank 0
410
m
m
Bank 1
420
400
Interface 450
n
DQ
Bank 2
430
m
m
Bank 3
440

FIG. 4

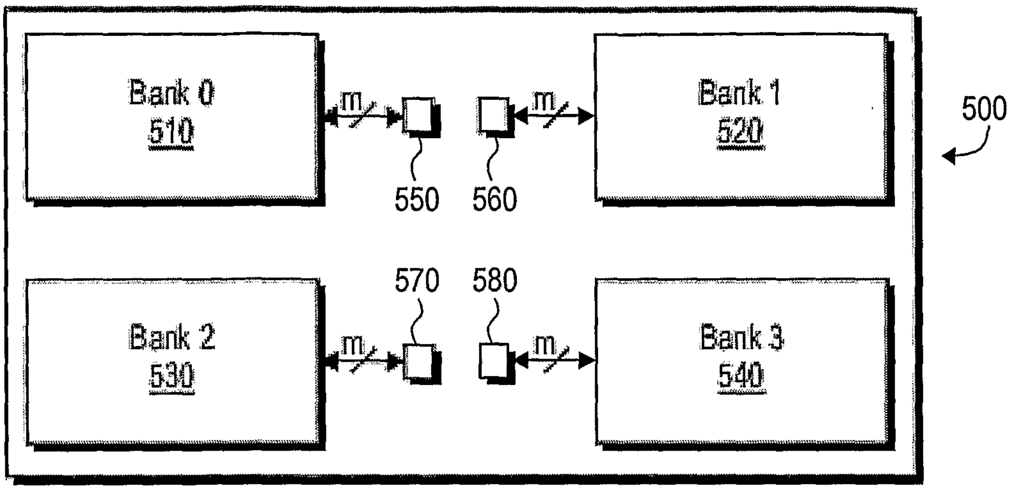

Bank 0
510
m
550
560
m
Bank 1
520
500
570
580
Bank 2
530
m
m
Bank 3
540

FIG. 5

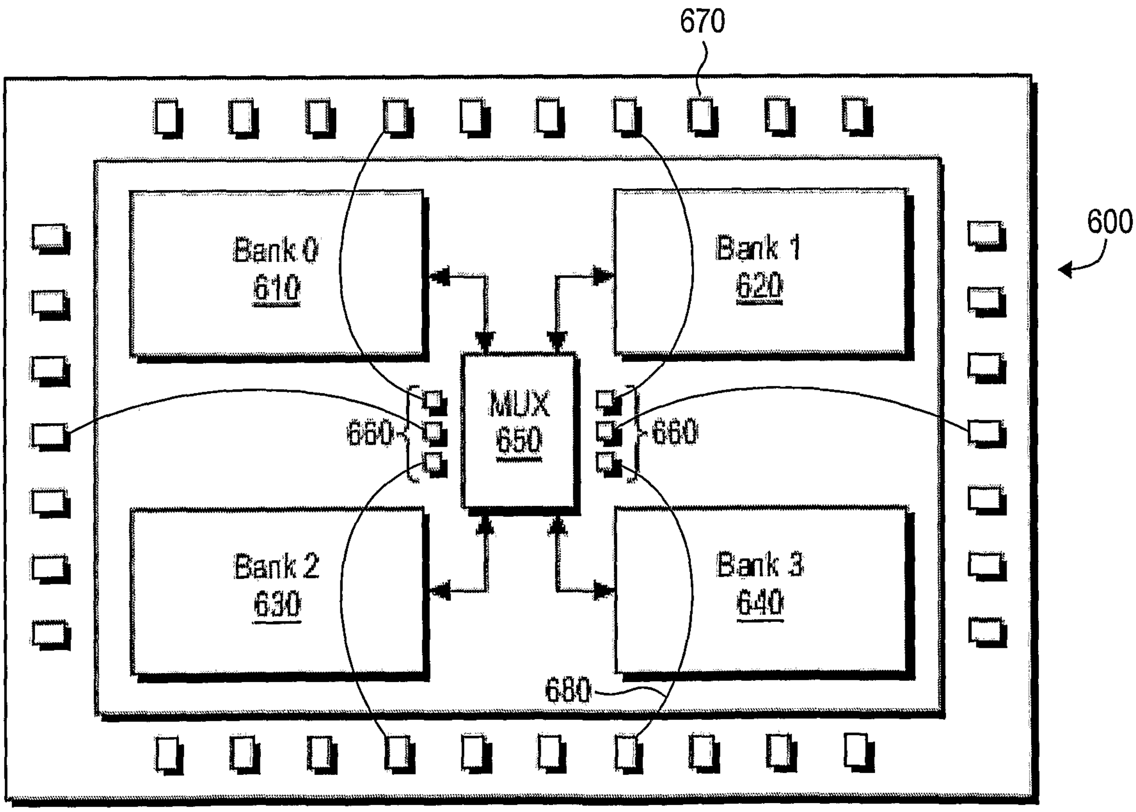

670
600
Bank 0
610
Bank 1
620
MUX
650
660
660
Bank 2
630
Bank 3
640
680

FIG. 6

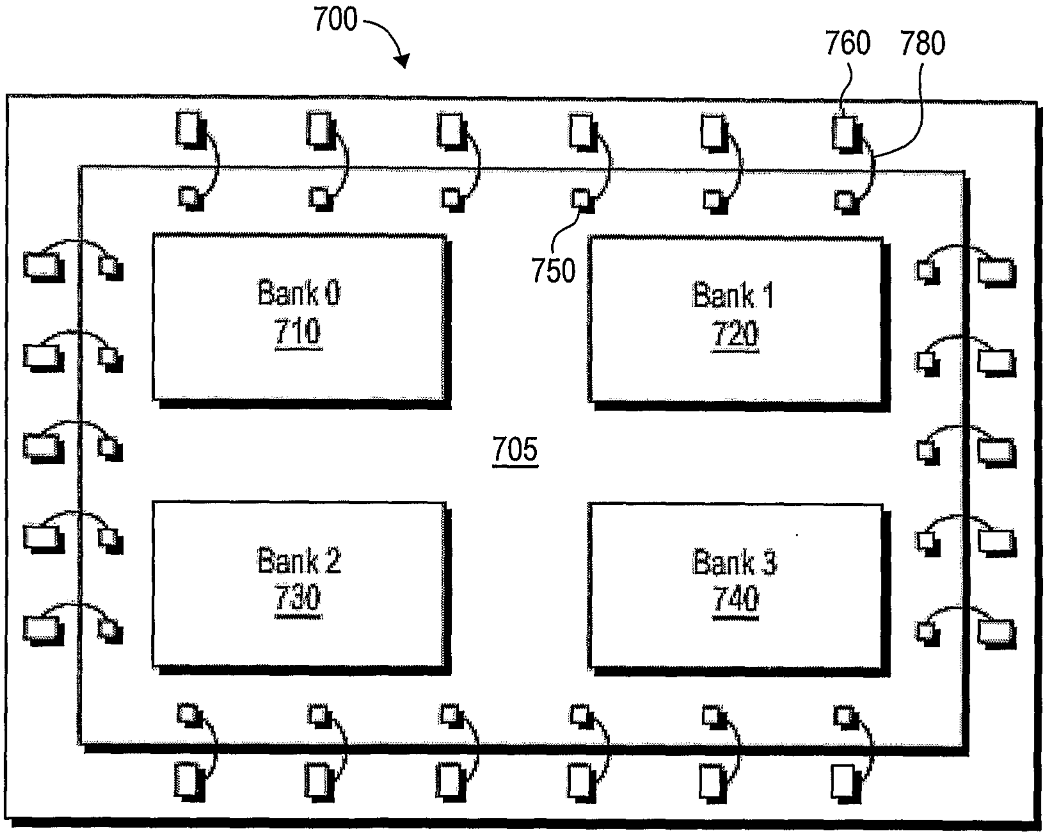



FIG. 7

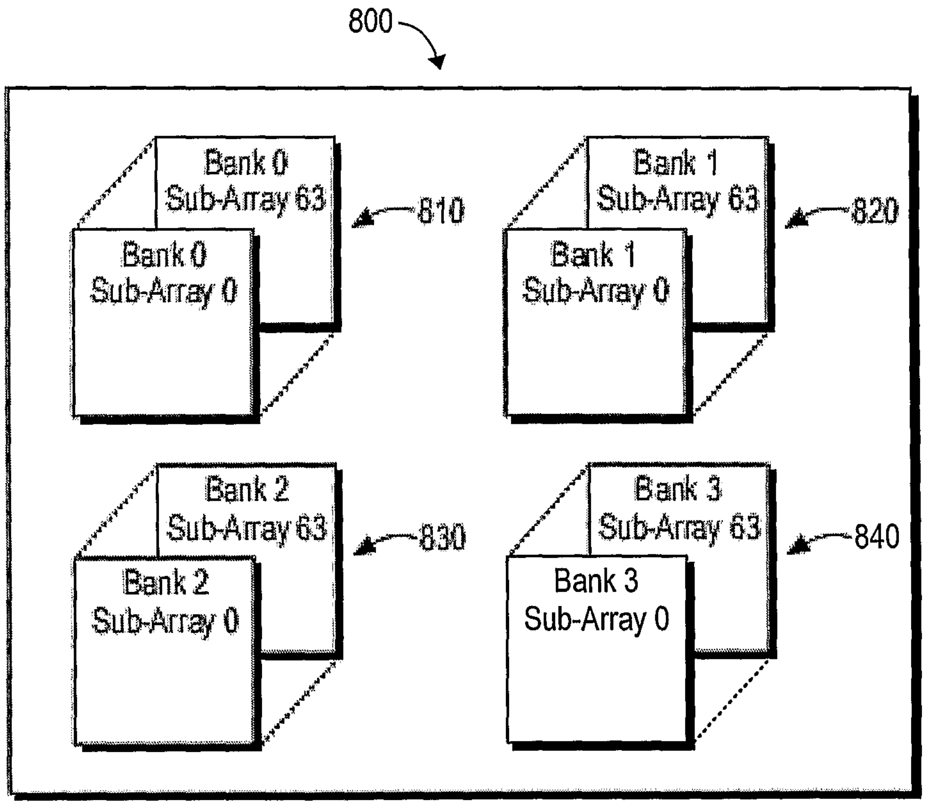

800
Bank 0
Sub-Array 63
810
Bank 0
Sub-Array 0
Bank 1
Sub-Array 63
820
Bank 1
Sub-Array 0
Bank 2
Sub-Array 63
830
Bank 2
Sub-Array 0
Bank 3
Sub-Array 63
840
Bank 3
Sub-Array 0

FIG. 8

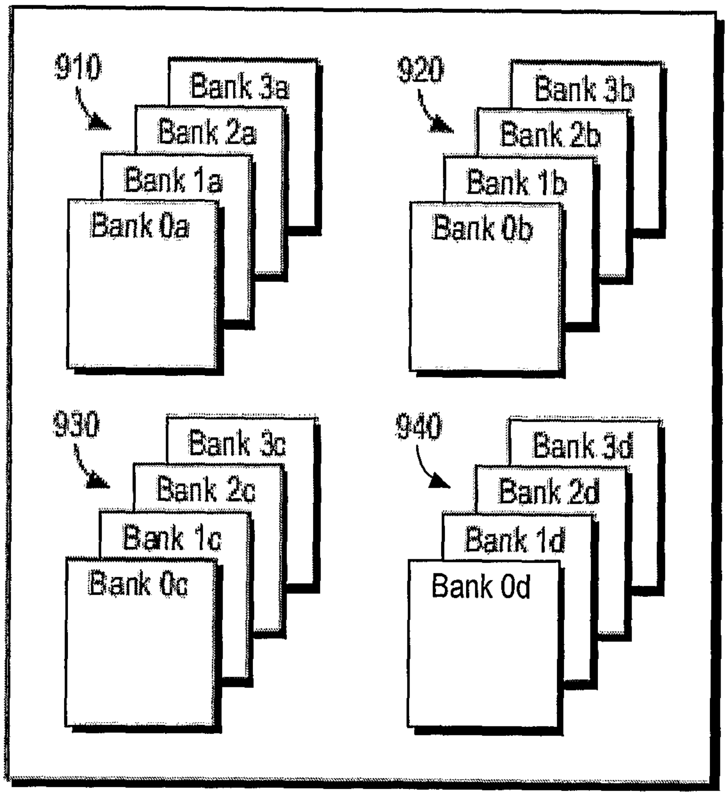
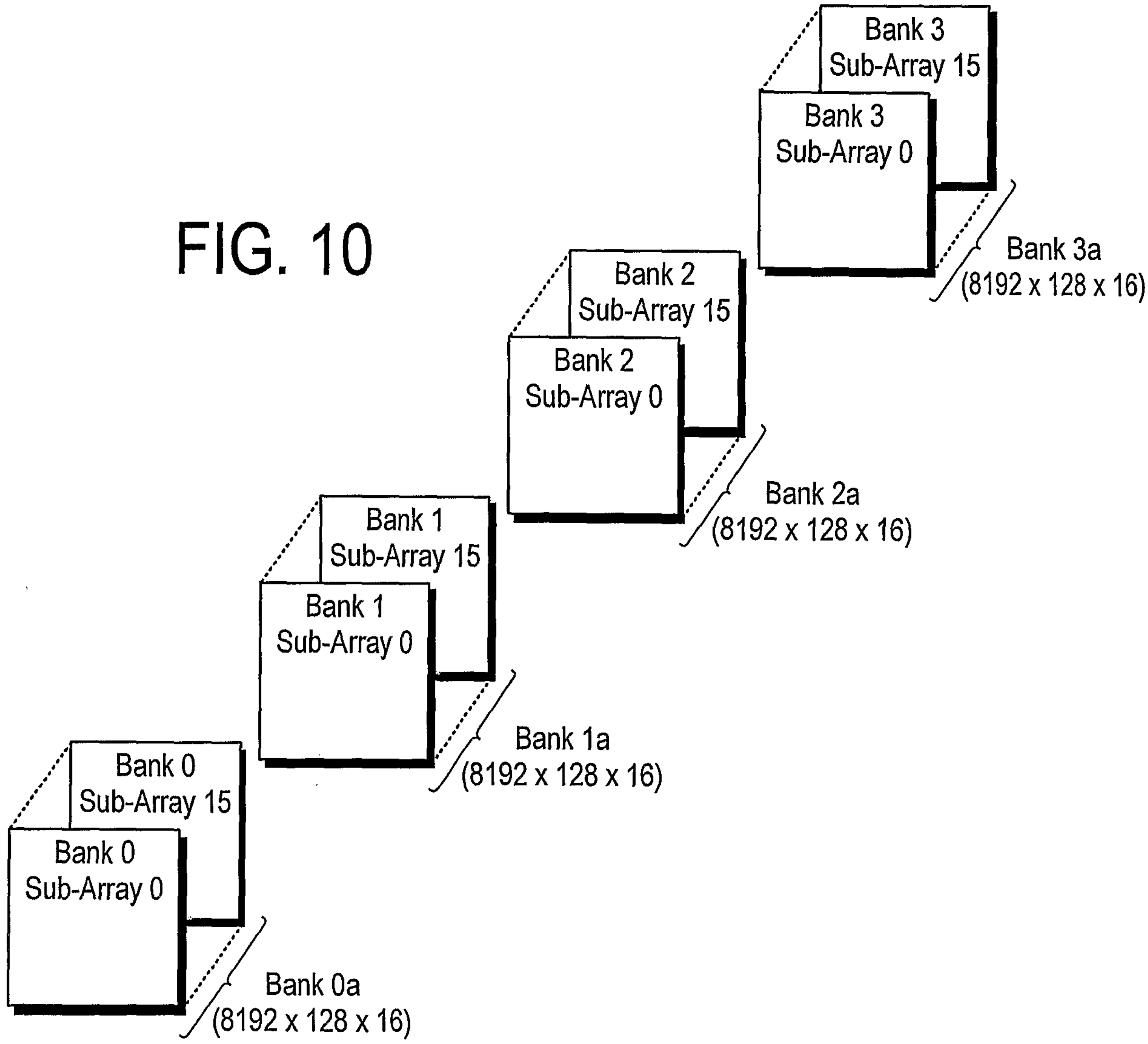

910
Bank 3a
Bank 2a
Bank 1a
Bank 0a
920
Bank 3b
Bank 2b
Bank 1b
Bank 0b
930
Bank 3c
Bank 2c
Bank 1c
Bank 0c
940
Bank 3d
Bank 2d
Bank 1d
Bank 0d

FIG. 9

FIG. 10


Bank 3
Sub-Array 15
Bank 3
Sub-Array 0
Bank 3a
(8192 x 128 x 16)
Bank 2
Sub-Array 15
Bank 2
Sub-Array 0
Bank 2a
(8192 x 128 x 16)
Bank 1
Sub-Array 15
Bank 1
Sub-Array 0
Bank 1a
(8192 x 128 x 16)
Bank 0
Sub-Array 15
Bank 0
Sub-Array 0
Bank 0a
(8192 x 128 x 16)

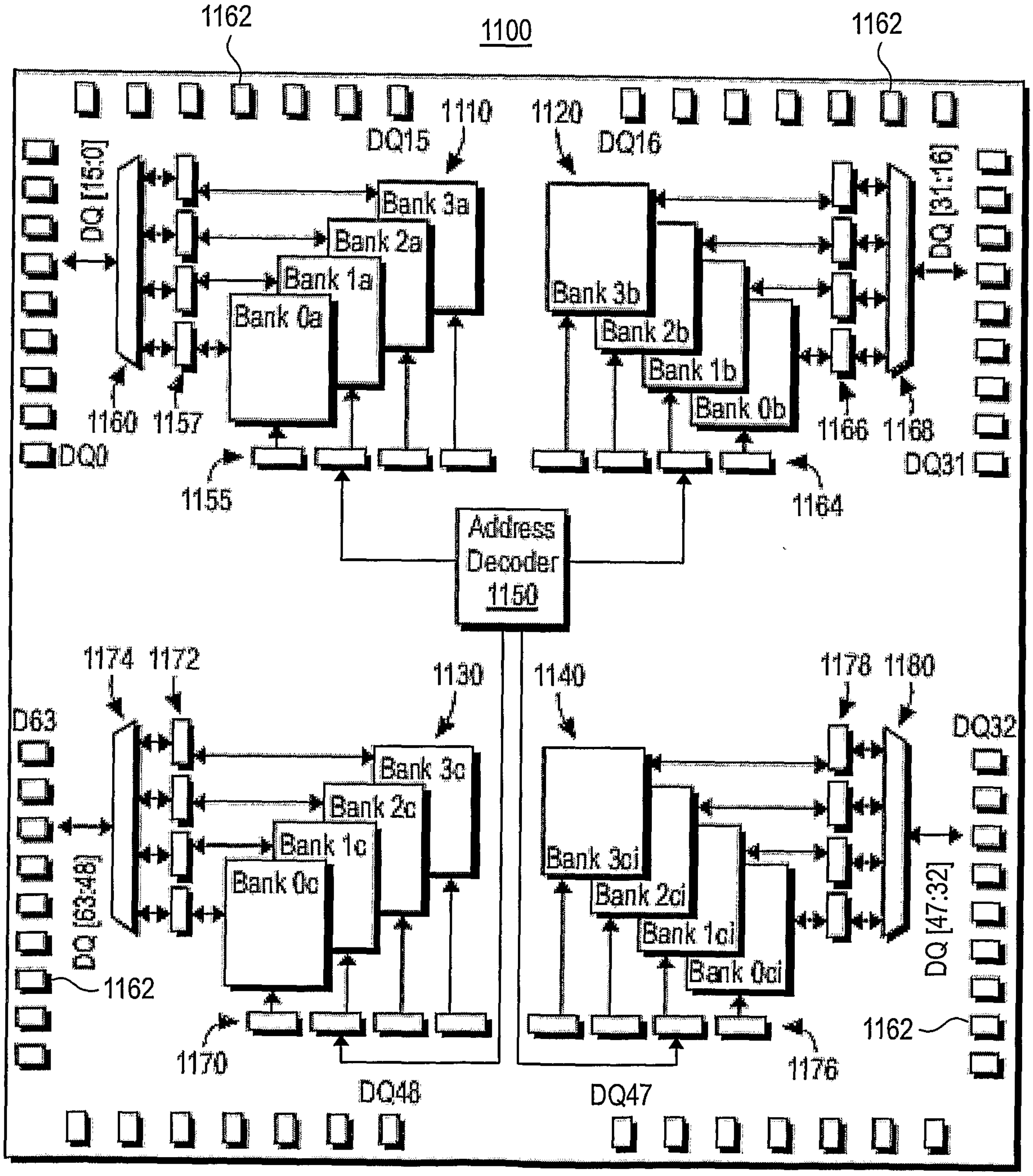

1100
1162
1162
1110
1120
DQ15
DQ16
DQ [15:0]
DQ [31:16]
Bank 3a
Bank 2a
Bank 1a
Bank 0a
Bank 3b
Bank 2b
Bank 1b
Bank 0b
1160
1157
1166
1168
DQ0
DQ31
1155
1164
Address Decoder 1150
1174
1172
1130
1140
1178
1180
D63
DQ32
Bank 3c
Bank 2c
Bank 1c
Bank 0c
Bank 3ci
Bank 2ci
Bank 1ci
Bank 0ci
DQ [63:48]
DQ [47:32]
1162
1162
1170
1176
DQ48
DQ47

FIG. 11

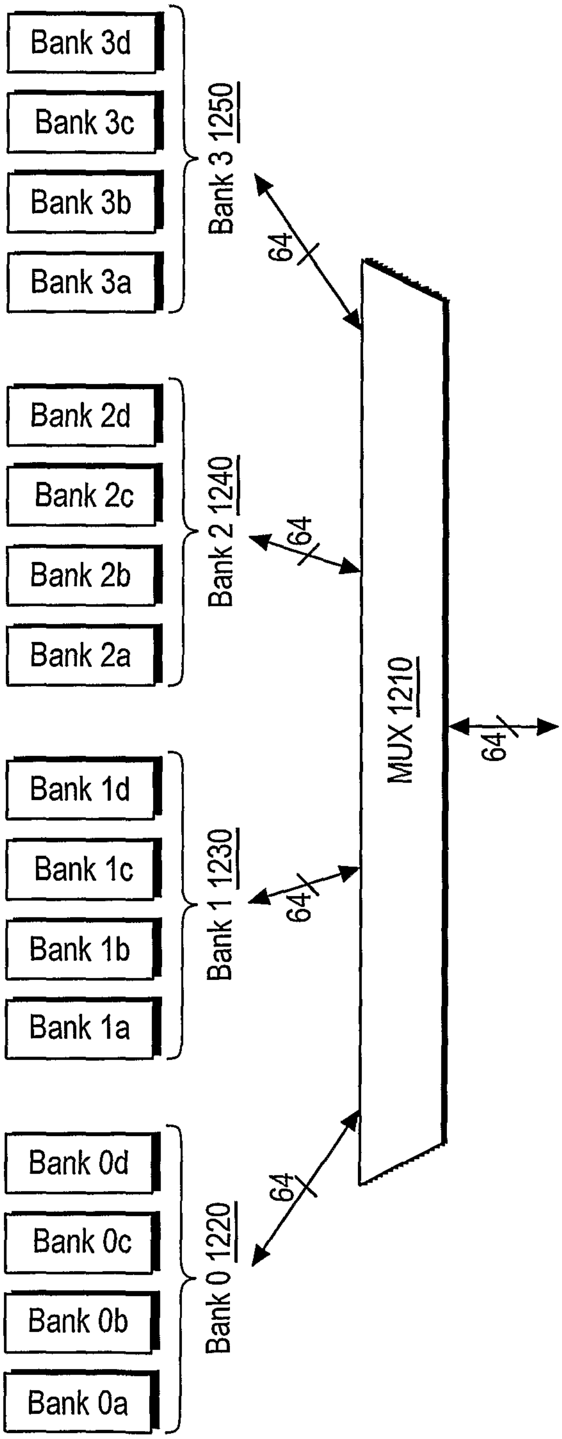

Bank 3d
Bank 3c
Bank 3b
Bank 3a
Bank 3 1250
Bank 2d
Bank 2c
Bank 2b
Bank 2a
Bank 2 1240
Bank 1d
Bank 1c
Bank 1b
Bank 1a
Bank 1 1230
Bank 0d
Bank 0c
Bank 0b
Bank 0a
Bank 0 1220
MUX 1210
64
64
64
64
64

FIG. 12

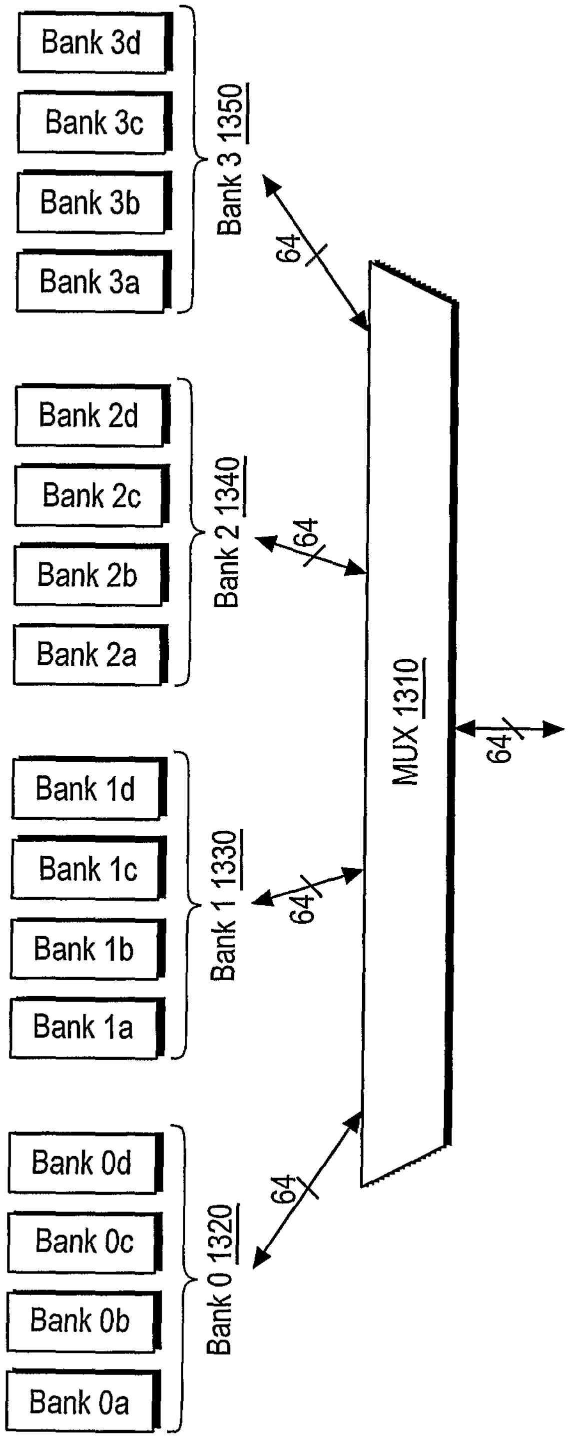

Bank 3d
Bank 3c
Bank 3b
Bank 3a
Bank 3 1350
Bank 2d
Bank 2c
Bank 2b
Bank 2a
Bank 2 1340
Bank 1d
Bank 1c
Bank 1b
Bank 1a
Bank 1 1330
Bank 0d
Bank 0c
Bank 0b
Bank 0a
Bank 0 1320
64
64
64
64
MUX 1310
64

FIG. 13

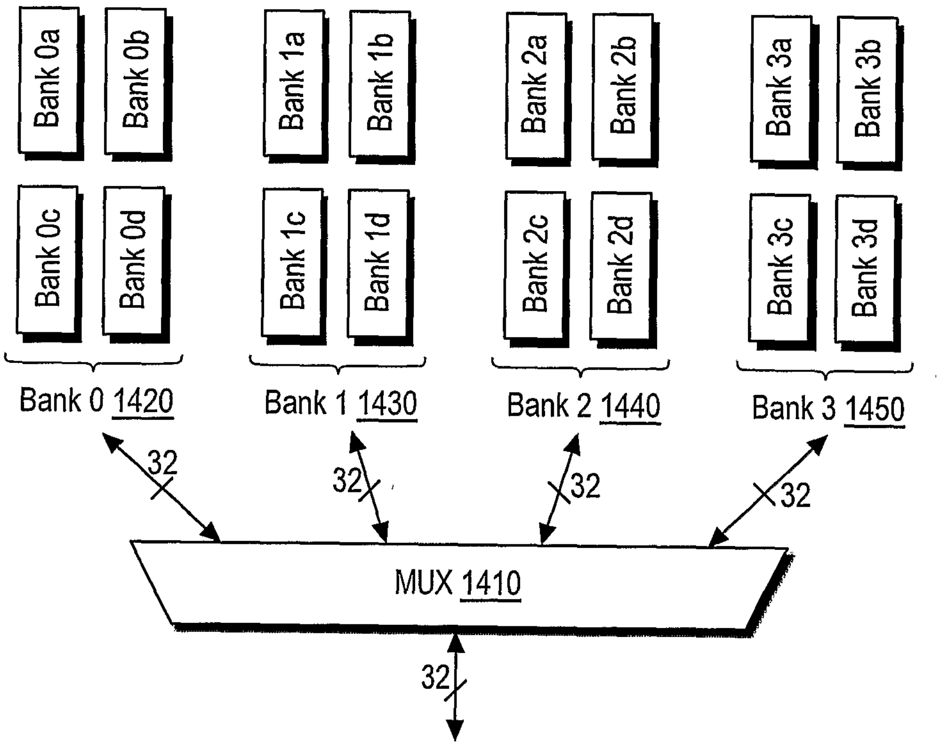

Bank 0a
Bank 0b
Bank 0c
Bank 0d
Bank 1a
Bank 1b
Bank 1c
Bank 1d
Bank 2a
Bank 2b
Bank 2c
Bank 2d
Bank 3a
Bank 3b
Bank 3c
Bank 3d
Bank 0 1420
Bank 1 1430
Bank 2 1440
Bank 3 1450
32
32
32
32
MUX 1410
32

FIG. 14

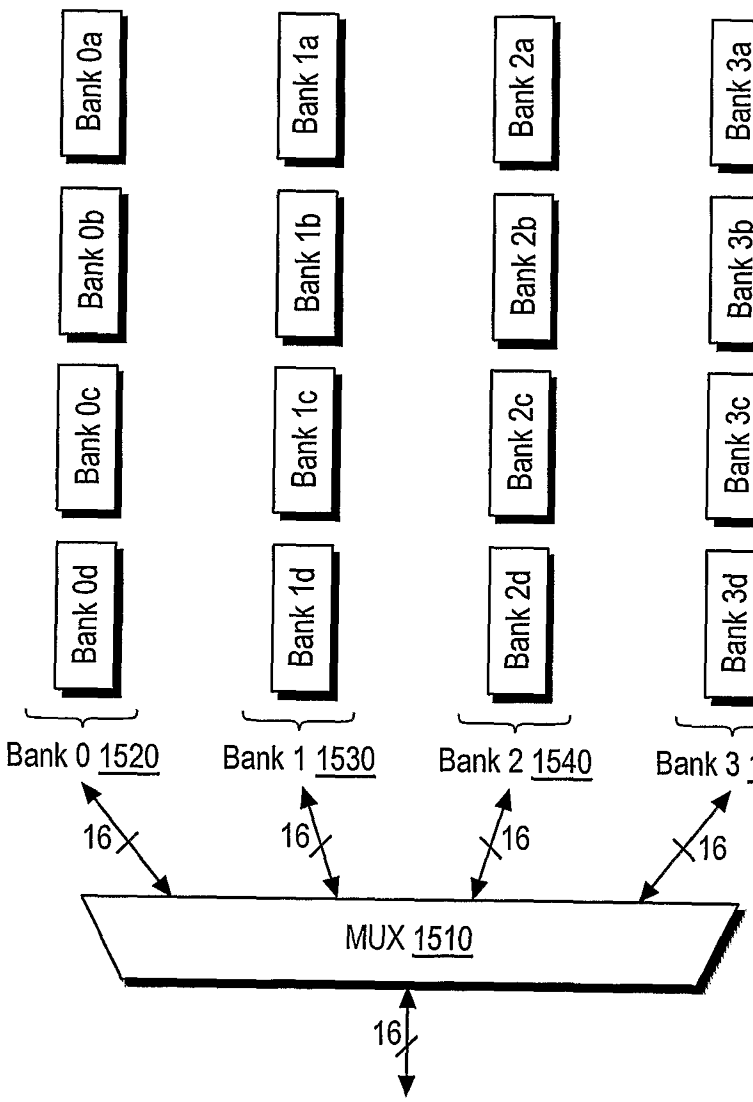
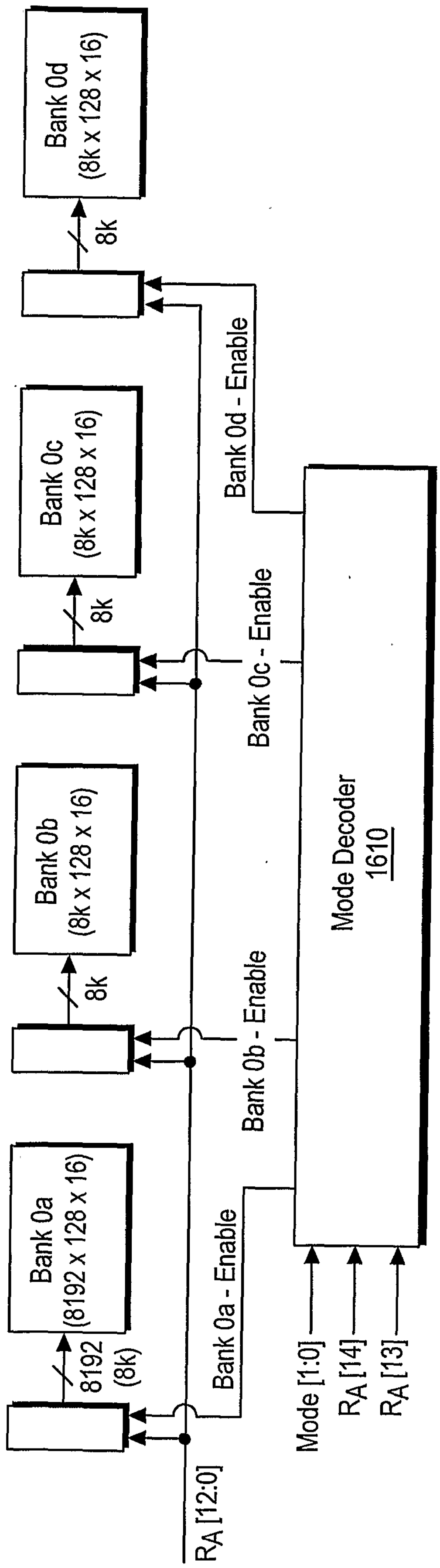

Bank 0a
Bank 0b
Bank 0c
Bank 0d
Bank 1a
Bank 1b
Bank 1c
Bank 1d
Bank 2a
Bank 2b
Bank 2c
Bank 2d
Bank 3a
Bank 3b
Bank 3c
Bank 3d
Bank 0 1520
Bank 1 1530
Bank 2 1540
Bank 3 1550
16
16
16
16
MUX 1510
16

FIG. 15



Bank 0d
(8k x 128 x 16)
8k
Bank 0d - Enable
Bank 0c
(8k x 128 x 16)
8k
Bank 0c - Enable
Bank 0b
(8k x 128 x 16)
8k
Bank 0b - Enable
Bank 0a
(8192 x 128 x 16)
8192
(8k)
Bank 0a - Enable
Mode Decoder
1610
Mode [1:0]
RA [14]
RA [13]
RA [12:0]


FIG. 16

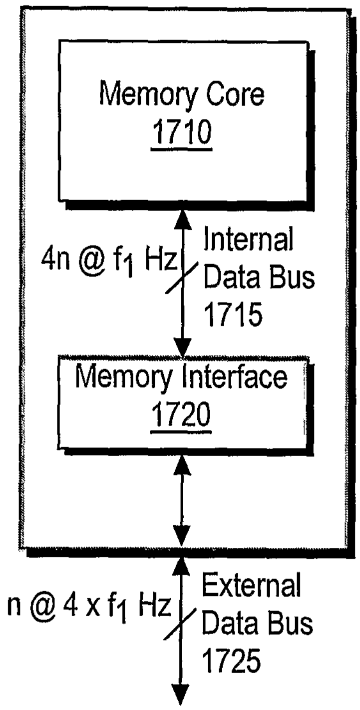



FIG. 17A

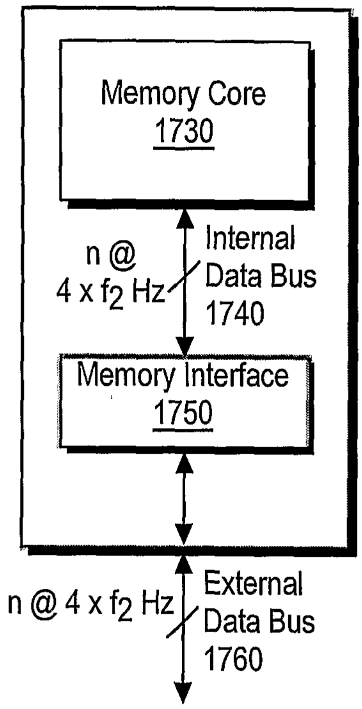
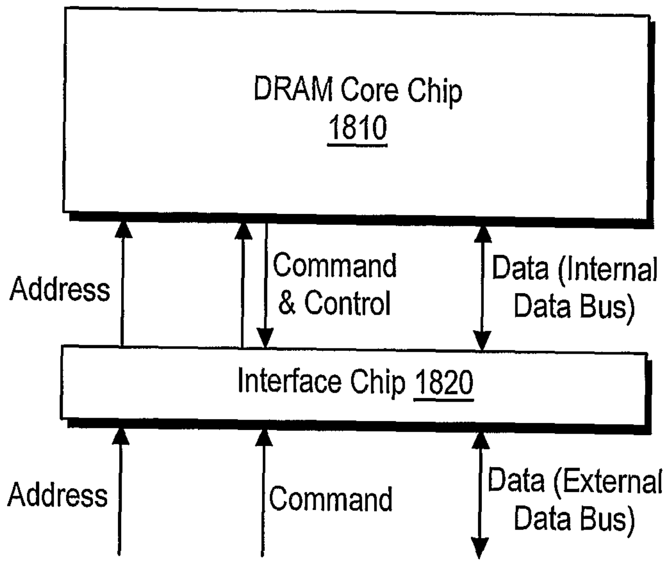



FIG. 17B



DRAM Core Chip
1810
Address
Command
& Control
Data (Internal
Data Bus)
Interface Chip 1820
Address
Command
Data (External
Data Bus)


FIG. 18

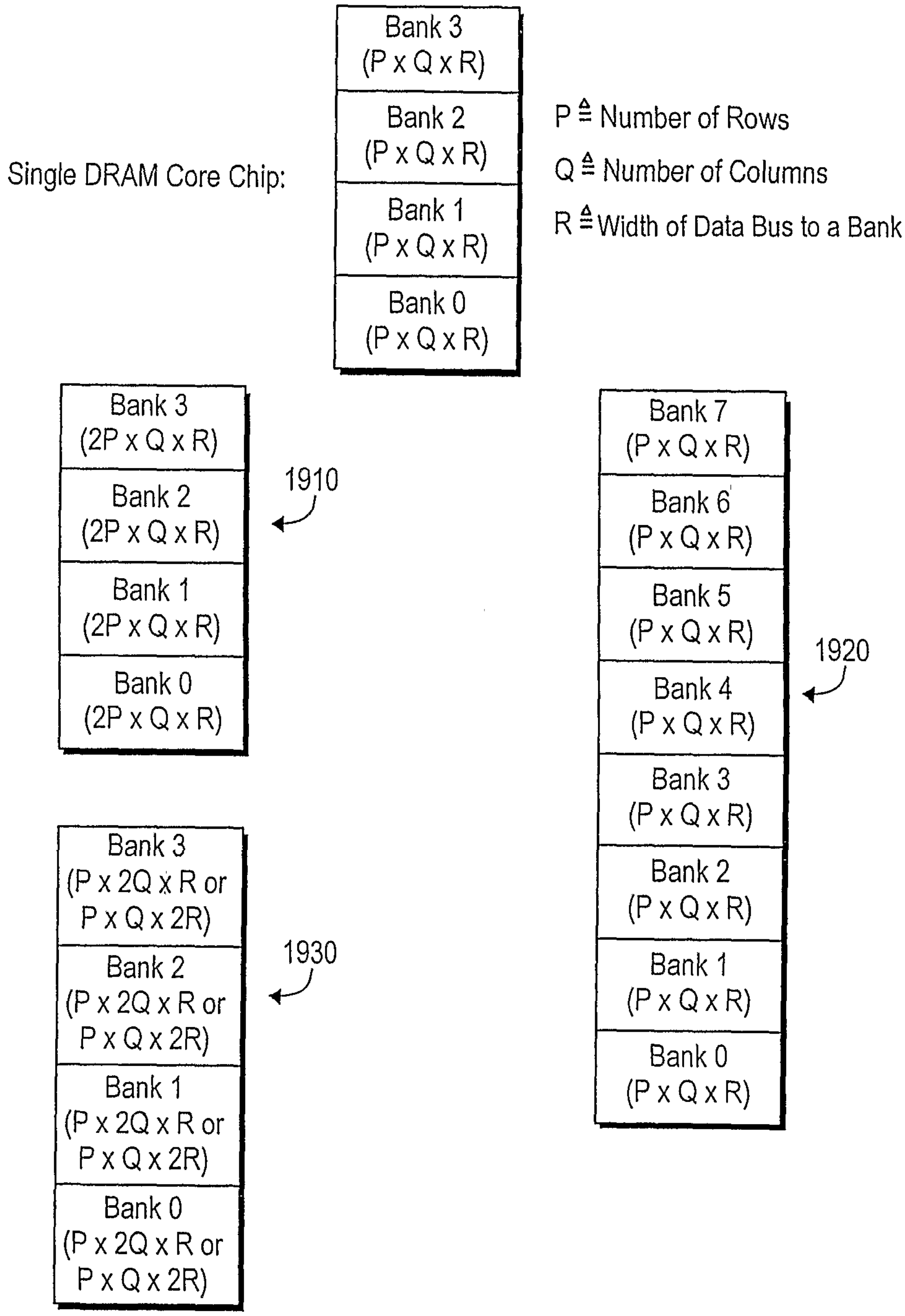

Single DRAM Core Chip:
Bank 3
(P x Q x R)
Bank 2
(P x Q x R)
Bank 1
(P x Q x R)
Bank 0
(P x Q x R)
P ≜ Number of Rows
Q ≜ Number of Columns
R ≜ Width of Data Bus to a Bank
Bank 3
(2P x Q x R)
Bank 2
(2P x Q x R)
Bank 1
(2P x Q x R)
Bank 0
(2P x Q x R)
1910
Bank 7
(P x Q x R)
Bank 6
(P x Q x R)
Bank 5
(P x Q x R)
Bank 4
(P x Q x R)
Bank 3
(P x Q x R)
Bank 2
(P x Q x R)
Bank 1
(P x Q x R)
Bank 0
(P x Q x R)
1920
Bank 3
(P x 2Q x R or
P x Q x 2R)
Bank 2
(P x 2Q x R or
P x Q x 2R)
Bank 1
(P x 2Q x R or
P x Q x 2R)
Bank 0
(P x 2Q x R or
P x Q x 2R)
1930

FIG. 19