Memory unit

- k address lines
- Read
- Write

- $2^k$ words
- n bits per word

- n data input lines
- n data output lines
<table>
<thead>
<tr>
<th>Memory address</th>
<th>Binary</th>
<th>Decimal</th>
<th>Memory contents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>000000000000</td>
<td>0</td>
<td>10110101 01011100</td>
</tr>
<tr>
<td></td>
<td>000000000001</td>
<td>1</td>
<td>10101011 10001001</td>
</tr>
<tr>
<td></td>
<td>000000000010</td>
<td>2</td>
<td>00001101 01000110</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>11111111101</td>
<td>1021</td>
<td>10011101 00010101</td>
</tr>
<tr>
<td></td>
<td>11111111110</td>
<td>1022</td>
<td>00001101 00011110</td>
</tr>
<tr>
<td></td>
<td>11111111111</td>
<td>1023</td>
<td>11011110 00100100</td>
</tr>
</tbody>
</table>
### TABLE 9-1
Control Inputs to a Memory Chip

<table>
<thead>
<tr>
<th>Chip select CS</th>
<th>Read/Write R/W</th>
<th>Memory operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>None</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Write to selected word</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Read from selected word</td>
</tr>
</tbody>
</table>
(a) Write cycle

(b) Read cycle
Select

RAM cell

B

\overline{B}

C

\overline{C}
(a) Symbol

RAM

16 × 1

Data input
Read/Write
Memory enable

(b) Block diagram

4-to-16 Decoder

Word select

RAM cell

Data input
Read/Write logic

Data output

RAM cell

Data in

Data out

Read/Write Bit select

Chip select

Data output
16 input data lines

Address 16

64K × 8 RAM
DATA
ADRS
CS
R/W

16 output data lines

16 input data lines

Chip select
Read/Write

64K × 8 RAM
DATA
ADRS
CS
R/W
Select

B
T
C
DRAM cell

(a)

To Pump

(b)

(c)

Select

B
D
Q
DRAM cell

model

C

(d)

(e)

(f)

(g)

(h)
Input/Output Logic

• Reference controller

• Refresh counter

• Row address register

• Row timing logic

• Column timing Logic

• Column address register

• Column decoder

• DRAM bit slice

• DRAM bit slice

Data in/
Data out

RAS

CAS

R/W

OE

Row address

Column address
(a) Write cycle

(b) Read cycle
### TABLE 9-2

**DRAM Types**

<table>
<thead>
<tr>
<th>Type</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast Page Mode DRAM</td>
<td>FPM DRAM</td>
<td>Takes advantage of the fact that, when a row is accessed, all of the row values are available to be read out. By changing the column address, data from different addresses can be read out without reapplying the row address and waiting for the delay associated with reading out the row cells to pass if the row portion of the addresses match.</td>
</tr>
<tr>
<td>Extended Data Output DRAM</td>
<td>EDO DRAM</td>
<td>Extends the length of time that the DRAM holds the data values on its output, permitting the CPU to perform other tasks during the access since it knows the data will still be available.</td>
</tr>
<tr>
<td>Synchronous DRAM</td>
<td>SDRAM</td>
<td>Operates with a clock rather than being asynchronous. This permits a tighter interaction between memory and CPU, since the CPU knows exactly when the data will be available. SDRAM also takes advantage of the row value availability and divides memory into distinct banks, permitting overlapped accesses.</td>
</tr>
<tr>
<td>Double Data Rate Synchronous DRAM</td>
<td>DDR SDRAM</td>
<td>The same as SDRAM except that data output is provided on both the negative and the positive clock edges.</td>
</tr>
<tr>
<td>Rambus DRAM</td>
<td>RDRAM</td>
<td>A proprietary technology that provides very high memory access rates using a relatively narrow bus.</td>
</tr>
<tr>
<td>Error-Correcting Code</td>
<td>ECC</td>
<td>May be applied to most of the DRAM types above to correct single bit data errors and often detect double errors.</td>
</tr>
</tbody>
</table>