Overview

DRAM stands for dynamic random access memory. Dynamic refers to the need to periodically refresh DRAM cells so that they can continue to retain the stored bit. Because of the small footprint of a DRAM cell, DRAM can be produced in large capacities. By packaging DRAM cells judiciously, DRAM memory can sustain large data rates. For these reasons, DRAM is used to implement the bulk of main memory.

DRAM Cell Design

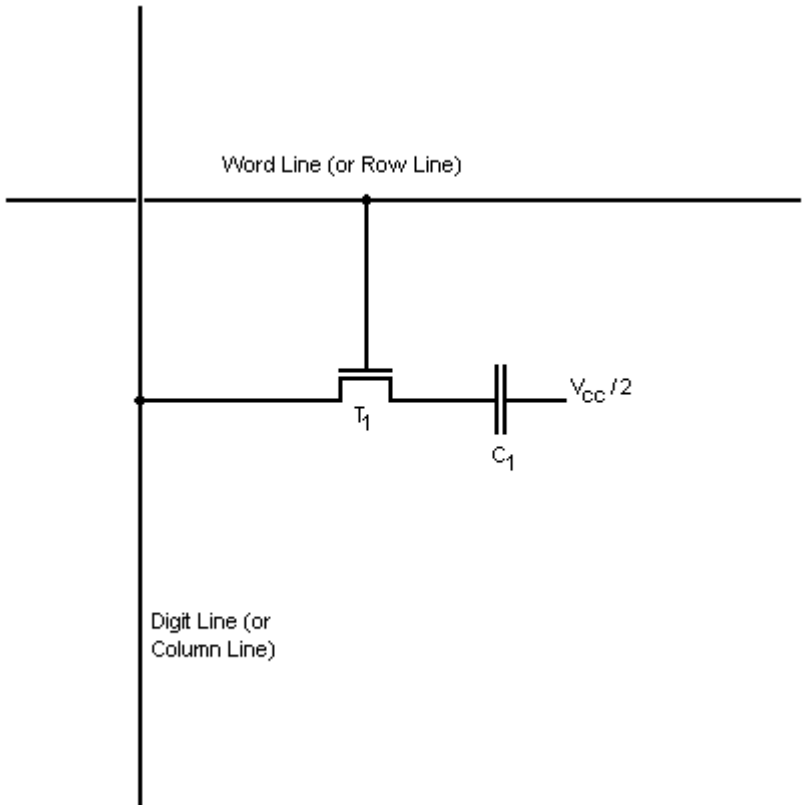

Figure 1: DRAM Cell

A DRAM cell consists of a capacitor connected by a pass transistor to the column line (or bit line or digit line). The column line (or digit line) is connected to a multitude of cells arranged in a column. The row line (or word line) is also connected to a multitude of cells, but arranged in a row. (See Figure 2.) If the row line is asserted, then the pass transistor T1 in Figure 1 is turned on and the capacitor C1 is connected to the column line.

The DRAM memory cell stores binary information in the form of a stored charge on the capacitor. The capacitor's common node is biased approximately at $V_{CC}/2$. The cell therefore contains a charge of $Q = \pm V_{CC}/2 \bullet C_{cell}$, if the capacitance of the capacitor is C_{cell}. The charge is $Q = +V_{CC}/2 \bullet C_{cell}$ if the cell stores a 1, otherwise the charge is $Q = -V_{CC}/2 \bullet C_{cell}$. Various leak currents will slowly remove the charge, making a refresh operation necessary.

If we turn on the pass transistor by asserting the row line, then the charge will spread over the column line, leading to a voltage change. The voltage change is given by (V_{signal} observed voltage change in the column line, C_{cell} the capacitance of the DRAM cell capacitor, and C_{line} the capacitance of the column line

$$V_{signal} = V_{cell} \bullet C_{cell} \bullet (C_{cell} + C_{line})^{-1}$$

For example, if V_{CC} is 3.3V, then V_{cell} is 1.65V. Typical values for the capacitances are C_{line} = 300fF and C_{cell} = 50fF. This leads to a signal strength of 235 mV. When a DRAM cell is accessed, it *shares its charge* with the column line.

DRAM Array Layout

The DRAM cells are arranged in a large rectangular structure with row lines controlling the gates of the pass transistors in all DRAM cells in a row and column lines collecting data from a large number of DRAM cells located in a column. (See Figure 2.) The length of the column increases the capacity of the DRAM array, but also increases the capacitance C_{line} and hence limits the signal strength.

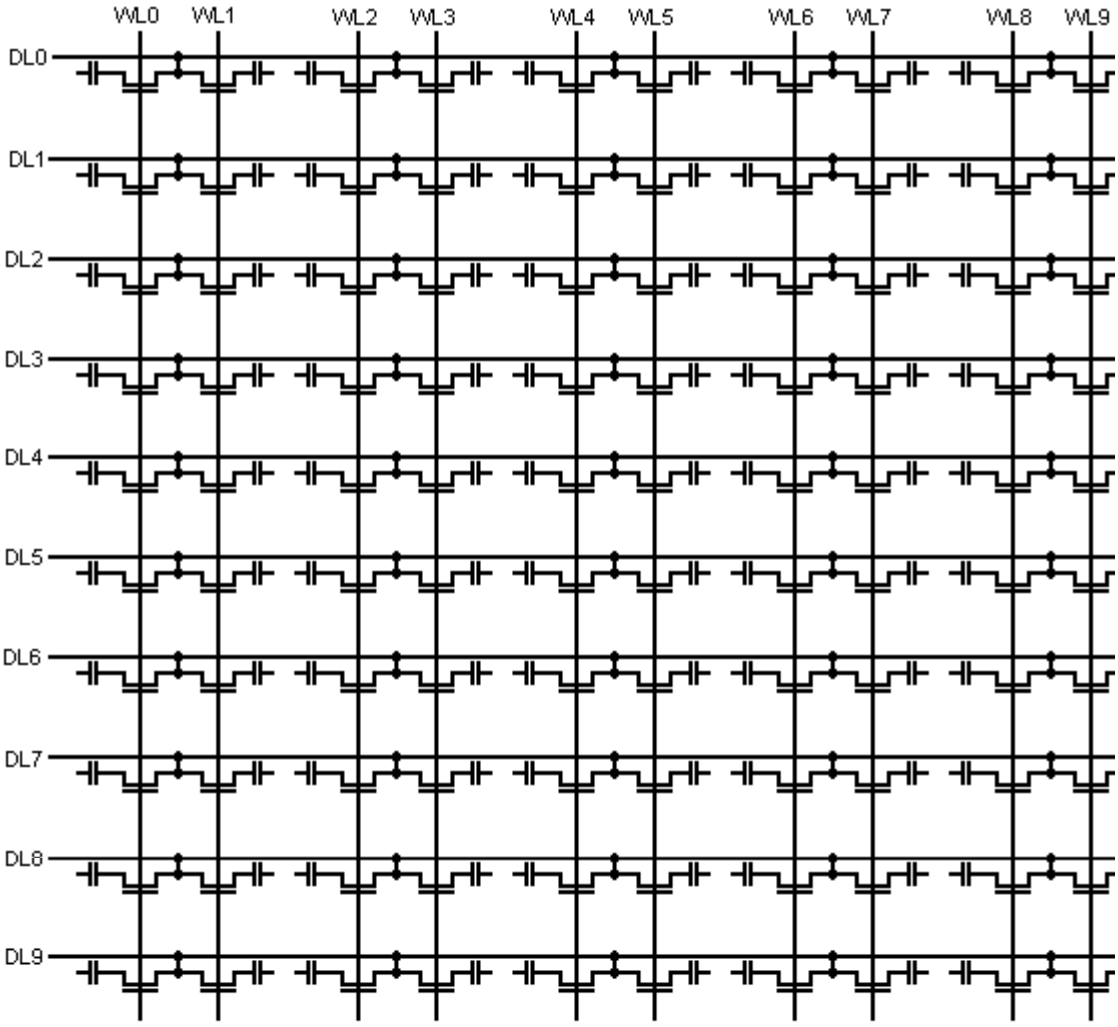

Figure 2: DRAM Array

.

Because the strength of the signal is small, DRAM design uses a reference line. Figure 3 shows the *open DRAM array layout,* where two DRAM arrays are located next to each other. Column lines at the same height are paired and gated into a sense amplifier. A sense amplifier pulls up the voltage differential between the two column lines. In a read operation, the other column line serves as a reference point.

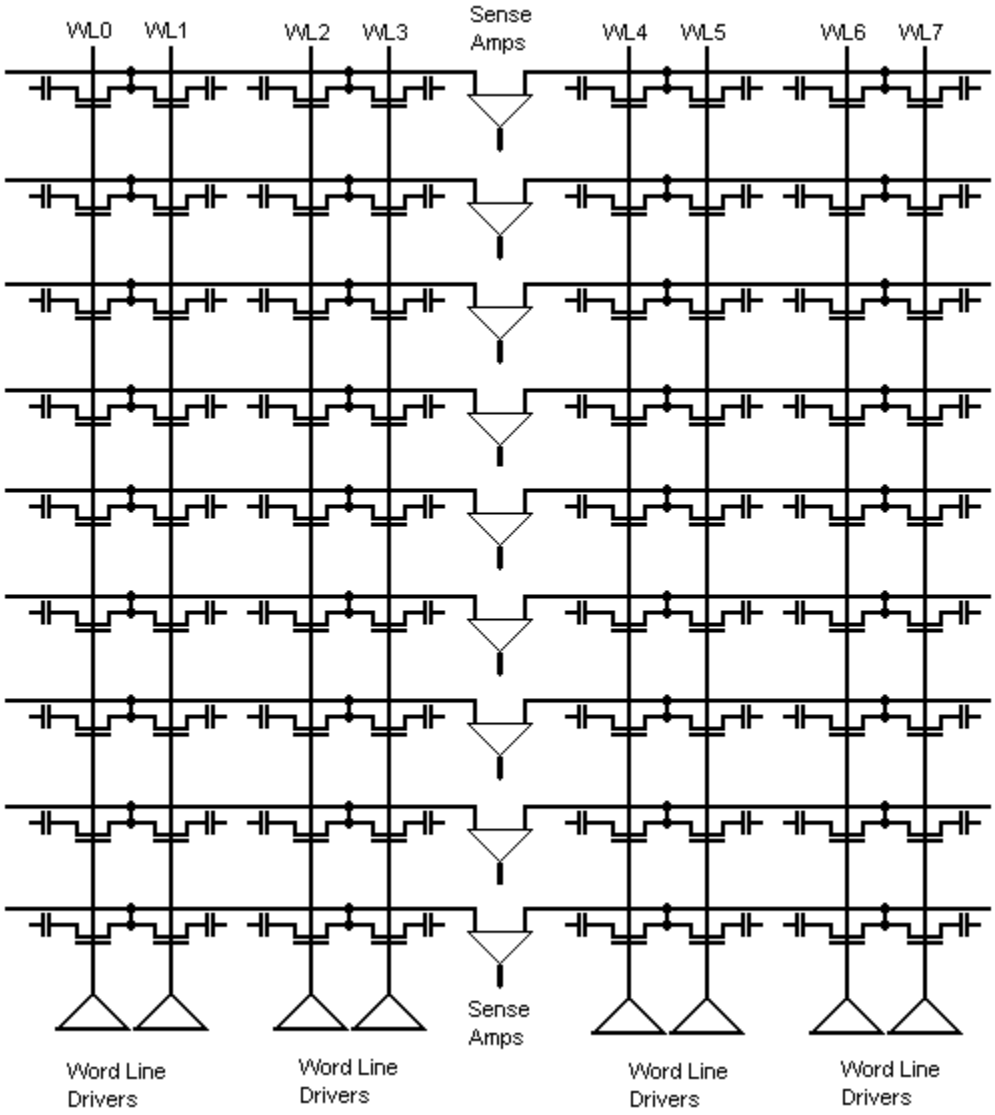

Figure 3: Open DRAM Array

Figure 4 shows a schematics of a second generation DRAM. The address is broken up into two pieces, a 10 bit row address (most significant digits of address) and a 12 bit column address (least significant digits of address). The address is strobed over in these two components. (This saves expensive pins since we now only need 12 address lines instead of 20.) To distinguish between the column and the write address, we use the CAS* and RAS* (column address strobe and row address strobe) signals. In addition, the chip has a Write* line to distinguish between read and write operations.

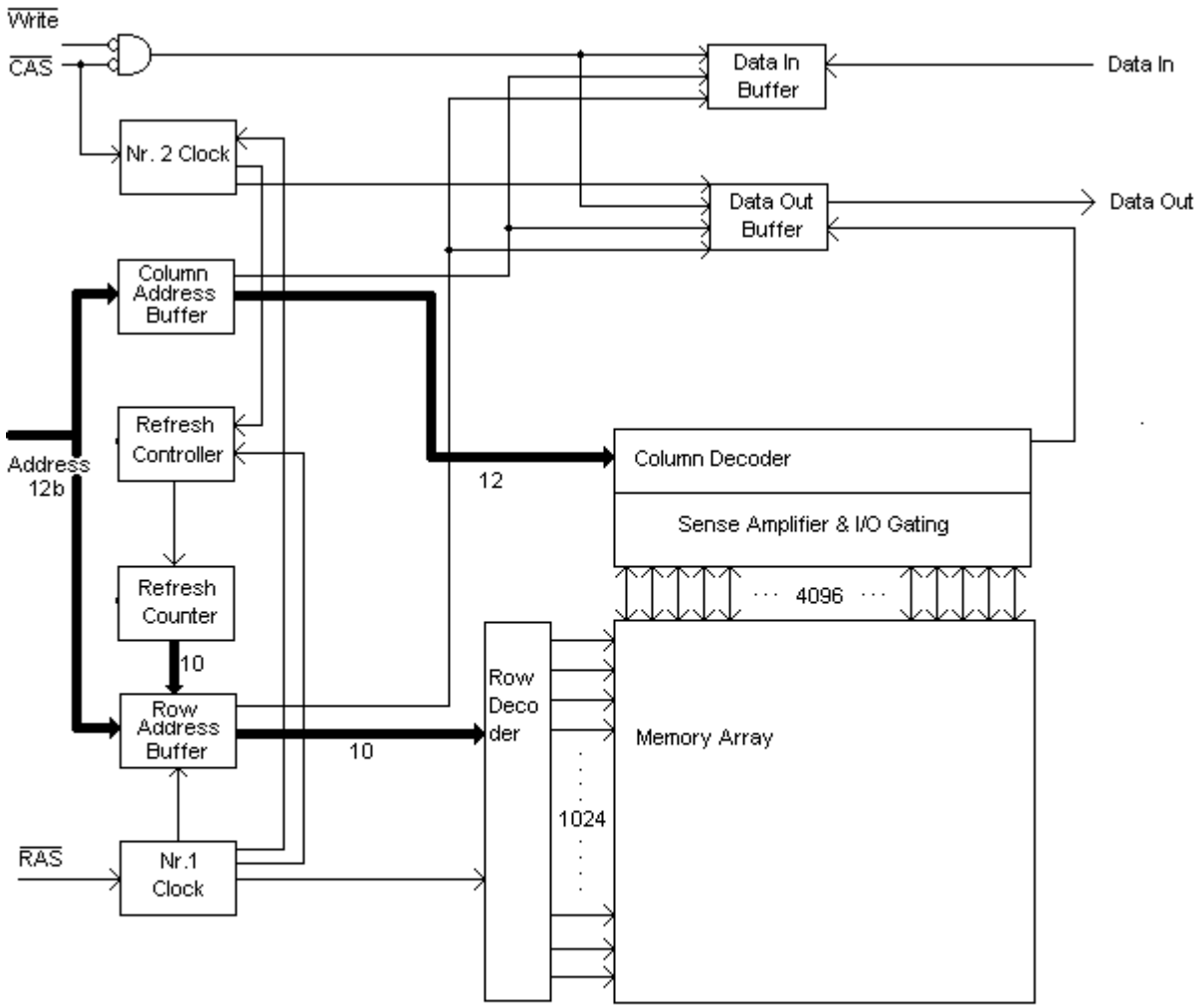

Figure 4: 4M * 1 DRAM (Siemens)

DRAM Operations

DRAM Read

Opening a row is a fundamental operation for read, write, and refresh operations.

1. Initially, both RAS* and CAS* are high. All column lines in the DRAM are ***precharged*** that is, driven to $V_{CC}/2$. All row lines are at GND level. This ensures that all pass transistors are off.
2. A valid row address is applied to the address pins of the DRAM and RAS* goes low. The row address is latched into the row address buffer on the falling edge of RAS* and decoded. The column lines are disconnected from the $V_{CC}/2$ bias and allowed to float. At this point, they are charged to this voltage of $V_{CC}/2$.
3. The decoded row address is applied to a row line driver. This forces one row line to high, thus

connecting a row of DRAM cells. The cell either lowers or raises the voltage in the column line it is connected to by V_{signal}.

4. After the cell has been accessed, sensing occurs. Sensing is essentially the amplification of the differential voltage between the two column lines D1 and D1* (see Figure 5). The P sense amplifier (the left side in Figure 4) and the N sense amplifier are generally fired sequentially. First, the N sense amplifier is fired by bringing NLAT* (N sense-amplifier latch) toward ground. As the voltage difference between NLAT and the column lines increases, the NMOS transistor whose gate is connected to the higher voltage column line begins to conduct. This conduction causes the low-voltage column to be brought to discharge towards NLAT* and finally to be brought to ground voltage. The other NMOS transistor will not conduct. Sometimes after the N sense amplifier has fired, ACT (for active pull-up) will be brought towards V_{CC} to activate the P snese amplifiers. As the low-voltage column line is close to ground, the corresponding PMOS transistor is driven into conduction. This charges the high-voltage column line towards ACT and ultimately towards V_{CC}. As a result of this operation, all column lines are either driven to high or to low according to the contents of the DRAM cell in the row.
5. The column address has been strobed into the column address buffer in the mean time. When CAS* falls, the column address is decoded and one of the sense amplifiers is connected to the data out buffer.
6. When RAS* is deasserted, the row line goes to low. As a consequence, the all DRAM cells in the row are now disconnected from the columnt line. Notice that all cells in the row have now been charged either to V_{cc} or to GND.

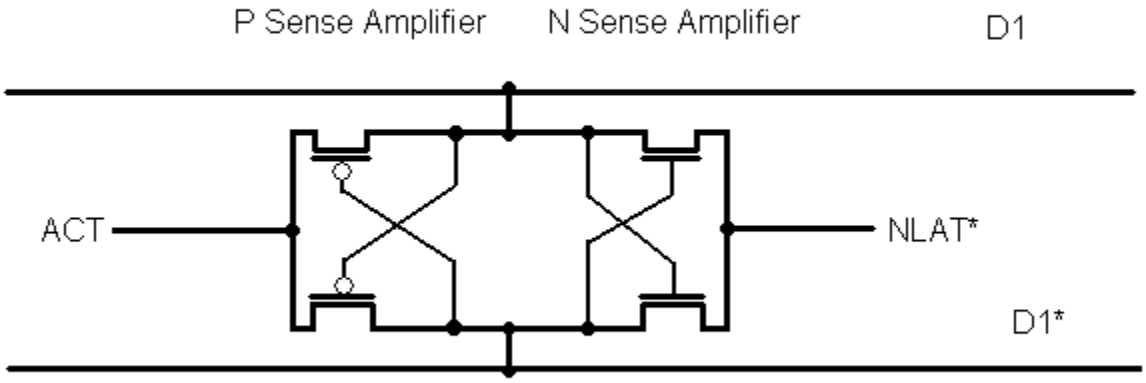

Figure 5: Sense Amplifier Schematic

DRAM Write

A DRAM write proceeds very much like a DRAM read. The main difference is the use of a separate write driver circuitry that determines the data that is placed in the cell. In most current DRAM designs, the write driver simply overdrives the sense amplifiers. In more detail

1. RAS* and CAS* are high. All Col. lines are precharged.
2. A valid row address is applied to the row address decoder and RAS* goes low. This enables the row address decoder so that a single row line (corresponding to the address) goes high. The connects all the cells in this row to the column lines.
3. The Col. lines are pulled up or down by the sense amplifiers according to the contents of the cell.
4. The datum is applied and the write driver enabled (because WRITE* is deasserted).

5. A valid column address is applied to the column address decoder and CAS* goes low. The write driver overdrives the sense amplifier selected by the column address decoder.
6. RAS* and CAS* go high again. The row line goes low and all cells are now disconnected from the column lines.

DRAM Refresh

The capacitor in each DRAM cell discharges slowly. At certain intervals, we need to ***recharge*** the DRAM cell. This is achieved by reading the cell. A read will place the contents of the cell on the column line, which is then pulled up to full level (GND or V_{CC}) by the sense amplifiers. When the row line is deasserted, all cells in the row have their contents restored at full charge / discharge level.

A refresh operation thus refreshes all the cells in the same row at once!

Early DRAM memories were refreshed under system control. Every so often, the system would issue a read request that would refresh a particular row. Nowadays, the DRAM chip contains a timer that allows it to refresh autonomously. Besides the timer, the main component is the refresh counter that contains the address of the row that needs to be refreshed. When a refresh operation is finished, then the counter is set to the next row in a cyclical manner.

The need to refresh amounts to using a certain (small) portion of the DRAM bandwidth.

Timings

Access to a DRAM row presupposes that the column lines are precharged. After each access, column lines need to be precharged. This is a major restraint on DRAM operations.

Advanced DRAM Designs

Page / Burst Mode

A DRAM page consists of the cells in a row. As a read operation places all the contents of this cell into the sense amplifiers, we can read within a page without having to precharge the column lines again. This is a major time savings.

A DRAM in page mode is still controlled by the RAS* and CAS* lines. Initially, we use RAS* to strobe in a row address. By asserting and deasserting CAS*, we strobe in column addresses. As long as we do not leave the page, accesses are now performed much faster, since we do not have to strobe in the row address and - most importantly - do not have to precharge before each access.

EDO - Hyperpage Mode

Hyperpage mode (a.k.a. extended data out) improved on previous DRAM page mode designs by storing the input in a latch. As a consequence, the result from a read was longer available, which allowed faster cycle times.

Synchronization

Historically, DRAM has been controlled asynchronously by the processor. This means that the processor puts addresses on the DRAM inputs and strobes them in using the RAS* and CAS* signals. The signals are held for the required minimum length of time during which time the DRAM executes the request. Because DRAM accesses are slow, the processor has to enter into a wait state.

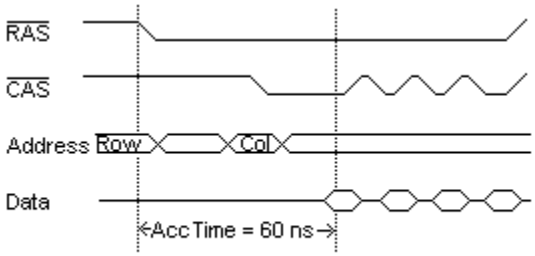

Figure 6: Asynchronous DRAM Timing in Nibble Mode

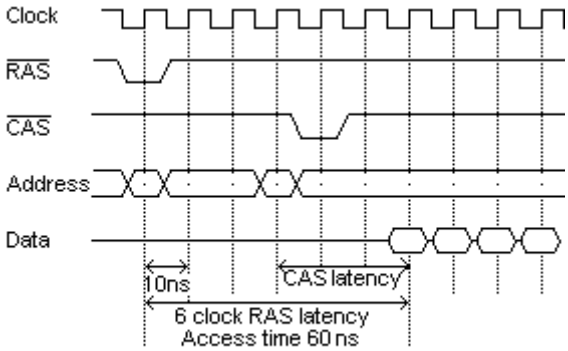

Figure 7: Synchronous DRAM Timing in Nibble Mode

Synchronization adds input and output latches to the DRAM and puts the memory device under the control of the clock. This alone can speed operations up, since there is no less need for signaling between processor and DRAM. Figure 6 shows the timing diagram of an asynchronous DRAM in nibble mode. The processor strobes in a row and a column address. 60 nsec after deassserting RAS* the first piece of data appears. By oscillating CAS* subsequent data appears. (The column address is automatically incremented after each access.) Figure 7 shows the same type of DRAM under synchronous control. The main difference is that no CAS* oscillation is needed in order to strobe out the additional data, which now appear once per clock cycle of 10 nsec, faster than under asynchronous control.

Banking

To stream out data faster than even in page/burst mode, DRAMs use a large number of memory arrays or banks. Consecutive accesses are then serviced by different banks. For example, we might employ two 1M·8 banks. The least significant bit of the address then selects between the two banks. If accesses use the two banks alternatively, then the operations can overlap, giving twice as fast data rates. A DRAM with banks has an additional internal comand to a bank, the **ACT** (activate) command that precharges the bank.

Pipelining

By pipelining the addresses, the **average** access time can be sped up. In this case, the input latch is used to store the incoming address, while the DRAM is still working on the previous command. The pipeline has three stages, the first for the input (address and possibly data), the second for the bank access, and the third for latching the output (for a read).

Prefetching

We can increase the speed of a synchronous DRAM by ***prefetching***. In this case more than one data word is fetched from the memory on each address cycle and transferred to a data selector on the output buffer. Multiple words of data can then be sequentially clocked out for each memory address.

DDR SDRAM and Rambus

DDR SDRAM (double data rate synchronous DRAM) began to appear in 1997 and offered a burst bandwidth of 2.1 GB/sec across a 64 bit data bus. DDR SDRAM uses a large input/output width of typically 32b, multiple banks (e.g. 4), prefetching, and pipelining. Commands are received on the rising edge of the clock, but data is available at both raising and falling clock edge, hence the name. DDR SRAM is designed to optimize the burst bandwidth.

Rambus Inc. has developed Rambus architecture and has licensed the design. It also offers a burst bandwidth of 2.1 GB/sec. However, by using a large number of internal banks, an internal SRAM write buffer, a very fast internal bus, and a sophisticated internal control, a Rambus DRAM has the lowest average random access latencies.

The Rambus architecture has three elements, the Rambus interface, the Rambus channel, and the RDRAM themselves. The Rambus interface is implemented on both the memory controller and the RDRAM devices on the channel. The Rambus channel is implemented via only 30 high-speed, low-voltage signals. Each channel supports up to 32 RDRAM. The channel has data and control bits moving in packets. Each packet takes four clock cycles to transmit.