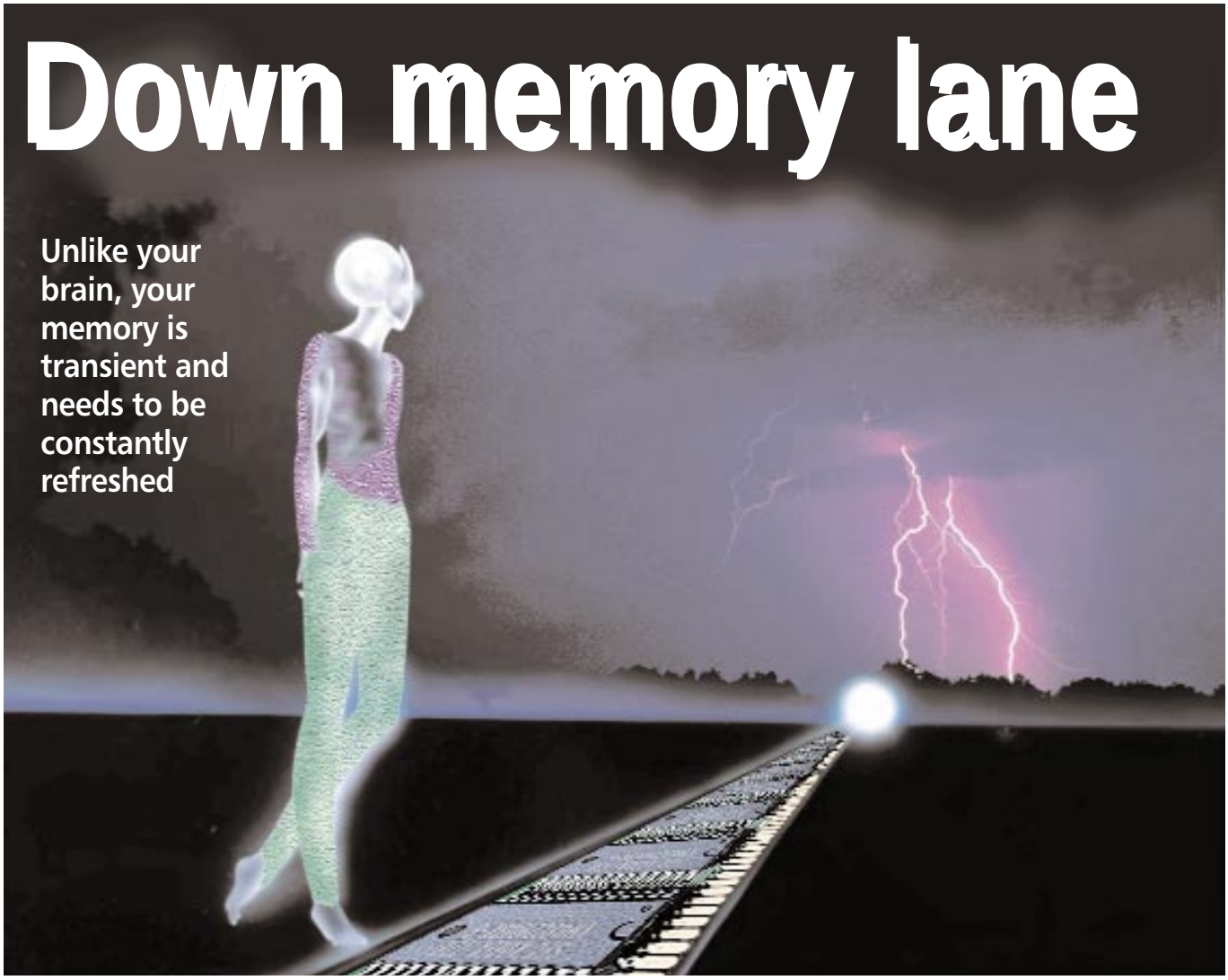
Down memory lane



Unlike your brain, your memory is transient and needs to be constantly refreshed

Only a couple of decades ago, computers had neither a hard disk drive or a floppy drive. You attached a cassette player to the computer and used audio tapes to store data. Not only was working with tapes unreliable and painfully slow, to get to some data in the middle of the tape, you had to put the tape in play mode and wait for 20 minutes or so before you got to the required location. Rewinding and forwarding were not allowed, as they would confuse the computer.

In contrast with this slow storage media was the computer's memory, which though limited in capacity and prohibitively expensive, was very fast and allowed you to locate data instantly. This remarkable form of storage media came to be called Random Access Memory (RAM).

More bits of information

This wonderful memory is just a collection of bits and a bit, in theory, is anything (such as a switch) that can be in only two states—'on' or 'off'. Memory, therefore, is a collection of switches. The electronic equivalent of a switch is the transistor which in concept resembles a push button switch. When you raise the voltage at the 'gate' (think of this as the button on a push button switch) the transistor is in 'on' state and allows the current to flow between 'source' and 'drain'. Remove the voltage and the transistor goes off (*Figure 1*).

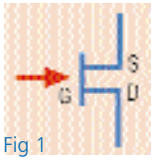


Fig 1

A transistor individually is not very useful, but starts to look interesting when paired with another transistor so that the two control each other. The resulting arrangement forms what electronic engineers call a 'flip-flop' circuit.

Figure 2 shows two transistors both with source connected to the 5V positive rail (called 'rail' because it connects a series of flip-flops), drain connected to the 0V 'ground' rail, and each one's gate connected to the other one's source.

Assume that the gate at Tr1 is turned on. Current flows through Tr1 faster than it can get through the resistor that connects Tr1 to the positive rail. This drops the voltage at point A to zero, effectively connecting point A to the ground rail. The gate at Tr2 connects to point A and is now also connected to the ground. As a result, Tr2 remains off, forcing current at point B to feed into the gate of Tr1; thereby keeping Tr1 'on' and taking us back to the starting state.

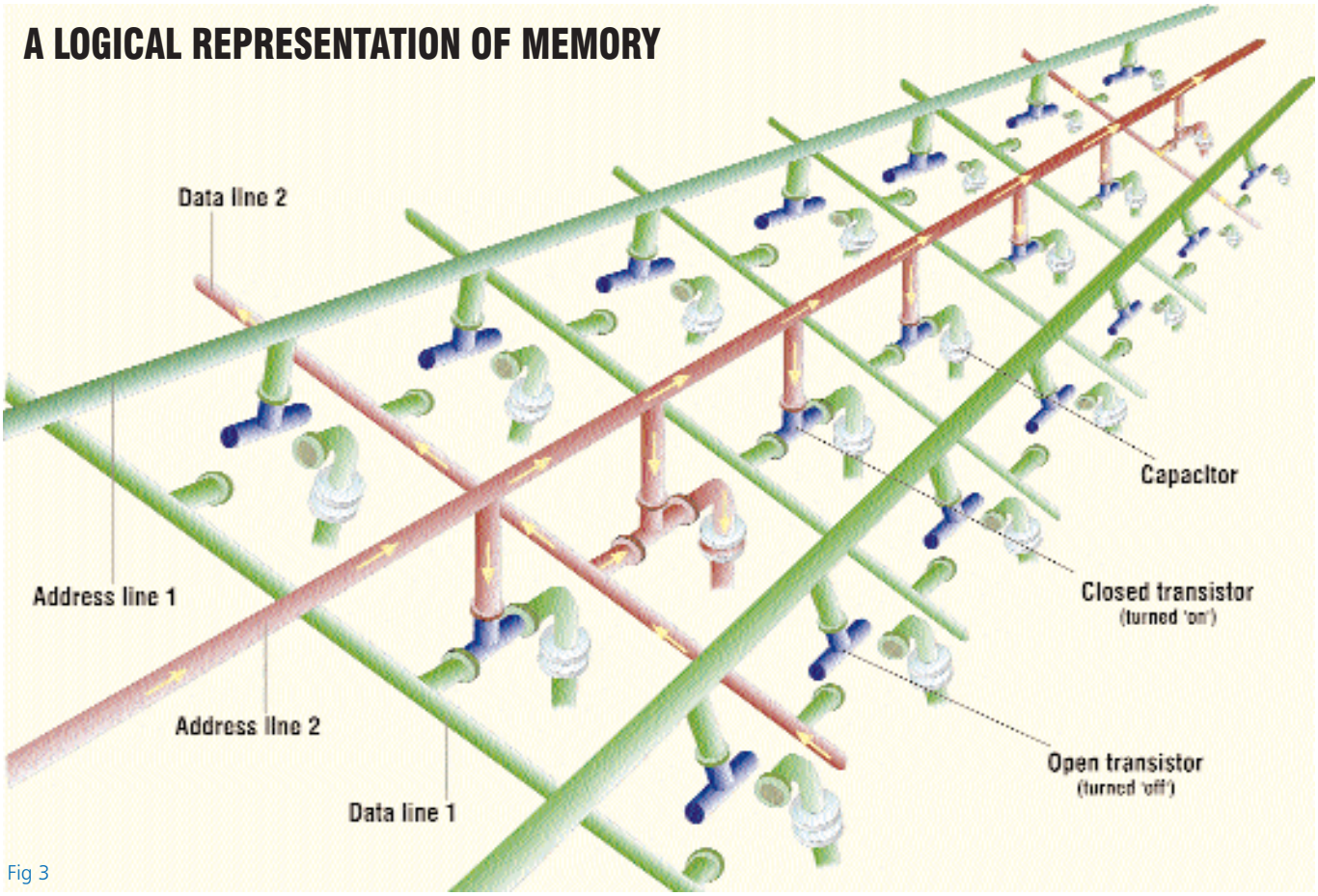


Fig 3

GRAPHICS: JAYA

To change the state of the cell from 'off' (Tr1 on, Tr2 off) to 'on' (Tr1 off, Tr2 on), a pulse of current is fed into point X. This pulse temporarily puts Tr2 in the 'on' state allowing the current to flow through Tr2. The point B drops to zero volts and Tr1 goes into the 'off' state. As a result the voltage at point A rises and the current is redirected to the gate of Tr2, thereby putting Tr2 in a permanently 'on' state. The flip-flop arrangement thus switches from an 'off' state to an 'on' state.

To reverse this state, the voltage at point X is dropped to zero (achieved by connecting point X to ground). This reduces the voltage at Tr2's gate to zero, thereby causing the chain reaction again where point B goes high, Tr1 turns on, point A goes low, and Tr2 therefore remains turned off.

Because the interlocking arrangement ensures that the two transistors retain the state they are put into, a flip-flop serves as a memory cell. You can know the state of the cell by looking at point Y. A high voltage there indicates that Tr2 is 'on', which means that the cell is in an 'on' state. A low voltage at Y indicates that Tr1 is on and Tr2 off, meaning that the cell is 'off'.

Each flip-flop can hold only one bit, which is just an eighth of a byte. It therefore takes millions of flip-flops to make up the many megabytes of memory that modern computers boast of. This type of memory is called Static RAM because of the tendency of the flip-flop to retain its 'on' or 'off' state, without external help. SRAM is fast and efficient on electricity consumption, but owing to the trouble involved in reproducing the intricate circuit, is also rather expensive. As a result, many other cheaper types of RAM have come to take its place. SRAM is now largely used in area where speed is a major concern such as in processor cache.

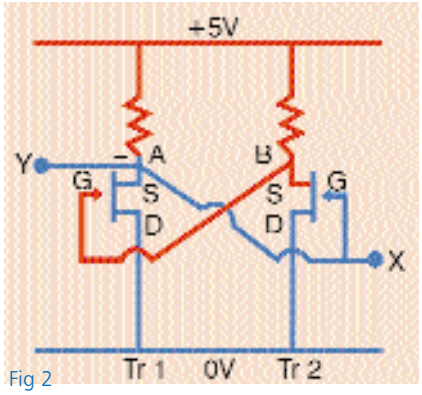


Fig 2

Forgetful memory

RAM that is currently available is called Dynamic RAM or DRAM. This type of memory is referred to as 'dynamic' because storage in the memory is temporary—the contents have to be refreshed every once in a while lest they should get lost or corrupted.

Dynamic RAM uses a combination of a transistor and a capacitor in place of the dual transistor flip-flop arrangement. A capacitor is a device that can 'hold' electric charge. In theory, you can use a capacitor as a memory cell. Feed it some charge and it attains a bit-value of '1'. Drain out the charge and the bit-value becomes '0'.

This ability of a capacitor and the

need to access one out of the many thousands of capacitors at a time results in an arrangement like that in Figure 3. The transistor attached to each capacitor functions as a switch to address the capacitor. It is not used for a bit-state storing mechanism.

The transistor-capacitor combination is arranged at every intersection point of a wire-grid pattern. The row wires on this grid are called address lines, while the column wires are called data lines. The gate of every transistor is connected to the address line, while the source connects to the data line and the drain connects to the capacitor. The other end of the capacitor is grounded.

To feed a charge into a capacitor, your computer's memory controller sends a pulse down the corresponding address line, and another pulse down the corresponding data line. The address line connects to the gate of all the transistors along the line. The pulse thus switches 'on' all the transistors along that particular line. When this happens, the pulse on the corresponding data line (which connects to the source of all the transistors) passes through the 'on' transistor into the capacitor, thereby charging it. The data pulse does not enter any other transistor along the data line, because in the absence of a pulse on its corresponding address line, every other transistor remains 'off'.

Reading from memory involves a similar procedure. A pulse is sent down the address line, and the data line is monitored to see if the capacitor releases a charge when the transistor turns 'on'. The presence of a charge indicates that the stored value is '1'. No charge means '0'.

Refreshing your memory

This memory storage system works fine except for a problem: a capacitor can retain data for barely a thousandth of a second. Even a read-attempt drains out the charge.

To get around this problem, the memory controller chip in your computer needs to continuously refresh memory contents—as often as many thousand times a second—for a computer to do anything useful with it, no matter what activity is taking place. Memory is refreshed once every two clock cycles. A clock cycle is a tick of your computer's internal clock.

HOW THE PROCESSOR SEES IT

The process of reading, writing and maintaining the contents of memory is completely hidden from software. An application running on your computer knows each memory location by its linear address. The memory controller converts this linear address to the physical address. The 8086 processor and its successor, the 8088, which powered the first IBM PC, had a 20-bit-wide address line. This meant that the processor and software running on it could address up to 1 megabyte of RAM (2^{20}=1 MB). However, a 20-bit address mechanism did not fit in well with the 16-bit wide data line, so the engineers at Intel decided to spread the 20 bits across two registers of 16 bits.

The first of these two 16-bit values was called the segment, the second the offset. To calculate the 20-bit linear address out of these two 16-bit values, the segment value was shifted left by 4 bits, and added to the offset value. The 16-bit wide data line meant that the processor could work with memory of 16 bits at a time.

The 80286 increased the size of the address line to 24 bits, making 16 MB of RAM possible. The 16-bit data line from the 8086 ancestry was retained. In 1987, the 80386 processor finally introduced the standard that exists today—a 32-bit address line and 32-bit data line, meaning up to 4 gigabytes of RAM and access to all that memory without having to do location address math first. The 32-bit 'flat' (no segment offset business) address mechanism was so good that, more than 10 years down the line, it has not yet run into problems. The 4 gigabyte memory limit is still a long way away for most computer users.

	E	E	2	C	
+		3	D	4	1
=	F	2	0	0	1

In hexadecimal (base 16), 4 bits make a digit. Therefore, to calculate the linear address from the segment and offset, the segment is shifted left by one digit (4 bits) and added to the offset.

The refresh procedure is very critical to the health of the memory's contents. Miss the refresh once and the contents get corrupted instantly. The actual refresh is performed by examining the contents of each memory cell, reading the charge and writing it back. Unbelievable as this sounds, it works efficiently and is used by all types of DRAM. Because the processor has to wait for every alternate cycle to be able to access memory contents, refreshing results in a performance loss against refresh-free Static RAM.

DRAM also has a problem with the required access time. Present day processors run at over 250 MHz or about 4 nanoseconds per cycle. DRAM access speeds average around 50 nanoseconds which is far slower than needed to keep up with the processor. If the processor cannot get the memory to keep up with it, it has to sit around doing nothing. This is known as the wait state and results in reducing the processing speed of your computer. The answer to this problem is to get faster RAM such as Static RAM. This is not practical for most computers, though, because of the higher costs involved.

Fortunately, there are cheaper solutions available. Caching is one such technique that uses a small amount of Static RAM in a special type of memory called cache memory. Processors tend to work with data in groups of bytes; therefore, by copying the entire group of bytes into the cache memory area and letting the processor work directly out of this area, system performance can be considerably improved. Generally speaking, the larger the size of the cache area, the better is the performance.

The memory controller is not alone in working overtime to keep memory contents updated. The processor also insists that it does something all the time, even if it is just processing instructions (from memory again) that tell the computer to wait for the user to respond. So the next time your computer is idle and your screensaver sleeps, think about the work the memory controller has to do to make things appear that way.

KIRAN JONNALAGADDA