Motherboards - The Designing Process
..:: Introduction ::.

Have you ever just looked at a motherboard and wondered how it was designed? How did the design engineers decide where to place certain chips, or components? What are these strange components placed all over the board? Why do motherboard manufacturers make a big deal out of multi-layer PCB's? Today we'll be taking a good look at what exactly goes into the process of a motherboard's design, from the planning stages to the finished product. Since this can be a very complex topic, I'm not going to be covering every last step, and I won't be going into incredible depth in areas that do not require it to be done. Not everyone out there is an aspiring CE or EE and has prior knowledge or the time to set aside and learn some of the more complex topics that I'll be discussing today. This article is based off of both continuing personal research, and aid from Rob Bruckner, a Senior Motherboard Design Engineer at Intel Corporation. This article is now in it's third revision as new information has been added, as well as modifications made to reflect upon current motherboards. The goal for the article remains the same, that you'll leave with a grown respect for the immense amount of thought and work that goes into the design process for a modern motherboard.

..:: Planning Groups ::..

In the infant stages of the motherboard design process, there are processes taken out that do not deal with the design itself, rather they deal with the end user will be wanting down the road. Remember, planning for processors, chipsets, and motherboards begins many months before they'll ever end up on the retail market. In this time, manufacturers will evaluate hat they feel their consumer base will want, and also how to make a distinctive product that will stand out against the rest of the pack.

The first steps in the design process for Intel start out with studies by various groups and individuals within the company detailing new chipset and integrated features. These groups are responsible for developing ideas and gathering feedback from the industry on what exactly it is that the consumers want in future products. Once this information has been gathered, Intel will decide which features it will want to implement in the future, and work then begins on next generation chipsets that will support these features. As these new products are developed, the information is passed on to the various motherboard manufacturers along with some reference designs and tools that they'll need to accomplish their work in time for the product launch.

Motherboards - The Designing Process

..:: Design Simulations ::..

The beginning stages of the actual motherboard design process starts out by running several complex simulations on either third party software, or in-house software depending on the manufacturer. These simulations are very important as they give the motherboard designers an overlay about proper trace routing, an especially important factor for the various system buses. These simulations determine the impendence of the traces and are used to establish the minimum and maximum length and width of the traces, along with the minimum allowable spacing between the traces. These factors all come into play for problems such as signal degradation, ground

bounce, EMI, etc.

Typically, today's printed circuit board designs aim for a trace impedance of roughly 60 Ohms, sometimes higher or lower depending on which bus the trace is going to be part of. A "normal" trace width that can be used for one of these high speed bus connections is roughly 5/1000 of an inch. When a motherboard designer starts work with the printed circuit board design, they'll work closely with the printed circuit board manufacturer in order to determine the optimal build, or "stack-up" of the various layers that will make up the motherboard. Motherboards today are designed with multiple layers in mind, each being responsible for a given task such as signaling, ground, or power distribution. Each of these layers is separated from each other by a "prepreg" layer which helps in determining the "static" impedance of the signaling traces. Below you'll find a simple example of a four layer PCB setup.

Layer 1 - Signal (Top Surface)

Prepreg (2116 Dielectric typically; ~4.5 mils thick)

Layer 2 - Copper Plane (Typically for Power Distribution)

Core (Thicker Dielectric than the Prepreg; Width adjusted to make the board thickness ~62 mils)

Layer 3 - Copper Plane (Typically for Ground)

Prepreg (2116 Dielectric typically; ~4.5 mils thick)

Layer 4 - Signal (Bottom Surface)

As mentioned in the above paragraph, the Prepreg actually helps to dictate the static impedance of the trace on the signaling layer/s. The impedance is determined by a combination of factors such as the Er, or dielectric constant, of the Prepreg, and the height of the trace above the copper layer, which is itself determined by the thickness of the Prepreg that is utilized.

When the motherboard designers and printed circuit board manufacturers attempt to find the best layer arrangement, they use something called "field solvers" to create a first pass at gaining the optimal layer arrangement for the targeted trace impedance. If a certain trace need to be, say around 7 mils wide, and the required impedance for the trace needs to be roughly 50 Ohms, the circuit board manufacturer will attempt to idealize their manufacturing process in order to meet these requirements and at the same time, achieve high yields for these impedances.

The impedances for the remaining traces are then solved by the circuit board manufacturer's own parameters for use with the motherboard. Now, instead of needing a 7 mil trace width with a 50 Ohm impedance, let's say now that we need a trace with an impedance of 60 Ohms instead. When the circuit board manufacturer runs their software, the solver will tell them that to achieve an impedance of 60 Ohms, a roughly 5 mil wide trace is needed.

If you take a look at the area around the processor socket, and where the bulk of the electrical components for the core voltage power supply are, you'll tend to notice that there are traces of varying widths and these traces can tend to be spaced out unequally. This is a great example of

how the circuit board manufacturers solver adjusts the width of each trace in order to meet the given impedance targets.

Now, if that's all that was needed, everything would be much easier than it is. When you have several traces near each other that are used as high speed interconnects for the various system busses, certain problems can arise that need to be addressed. If you were only dealing with a single trace running over one of the copper layers, the impedance of the trace would tend to stay near a given level and not vary much, hence the term "static impedance." However, when you're talking about several, tightly packed traces some problems can arise. One of the major problems is that you'll end up with cross-talk. When cross-talk arises, two things tend to happen. First, a high speed edge on one trace can "couple" itself with the signal from a neighboring trace which can cause distortion in that signal, something that obviously needs to be avoided. Secondly, depending on how the trace signal switches in comparison to the neighbor traces, you'll end up with either something called "even-mode" or "odd-mode" cross-talk. Each of these can be explained rather easily with an example of a three trace system.

In order to figure out if we're experiencing odd or even-mode cross-talk, we need to pay attention to what the middle trace is doing in comparison to the neighboring traces. If this middle trace is switching from high to low, while the two outside traces switch from low to high, then you have an odd-mode cross-talk situation. Even-mode cross-talk is when identical transitions happen. Here the center trace experiences a transition from low to high, while the neighboring traces also switch from low to high.

In the end, odd-mode and even-mode cross-talk can actually change the effective impedance of a trace. That 60 Ohm trace that you needed has now experienced a change in impedance due to the neighboring traces, and the change itself is dynamic which means the trace impedance will vary depending on the neighboring traces. This is where the simulations once again come into play. These simulations will find the optimal spacing between the various traces and the trace widths in order to deal with the potential for cross-talk. This is the main reason why you'll notice that some traces are farther apart than others, to keep the impedance target where it should be for the traces. Needless to say, this is a very complex and complicated process.

..:: Design Simulations - Layering Importance ::..

The layering layout of the motherboard is also very important, especially when it comes to servicing the various system busses and other high speed transmission lines. Each of these transmission lines requires something called a "return path." As current, a signal, travels down one of the high speed interconnect wires, a "return current" must flow in the opposite direction of the signal on the copper plane that is located directly under the interconnect. An easier way to understand this would be to think of the return current as a way to complete the circuit full circle. The key to this is for the designer to try to achieve the most efficient return path for this return current.

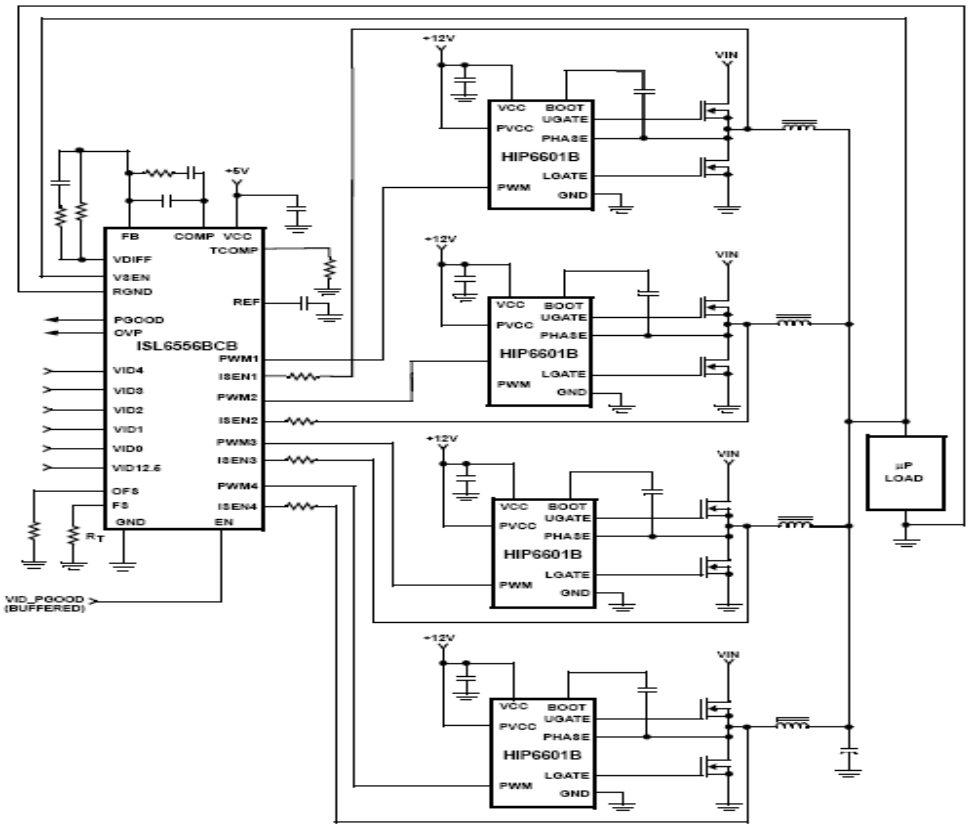
In order to achieve the most efficient return current, the designer must do two things. First off, the designer must attempt to maintain a single constant return path as much as possible. Secondly, the designer must make sure that both the source of the signal and the destination of the signal both tie into the return plane's voltage rail. In order to illustrate an example of this, let's discuss the 1.80V DDR-II system bus hat the new chipsets from Intel, and other upcoming manufacturers will utilize.

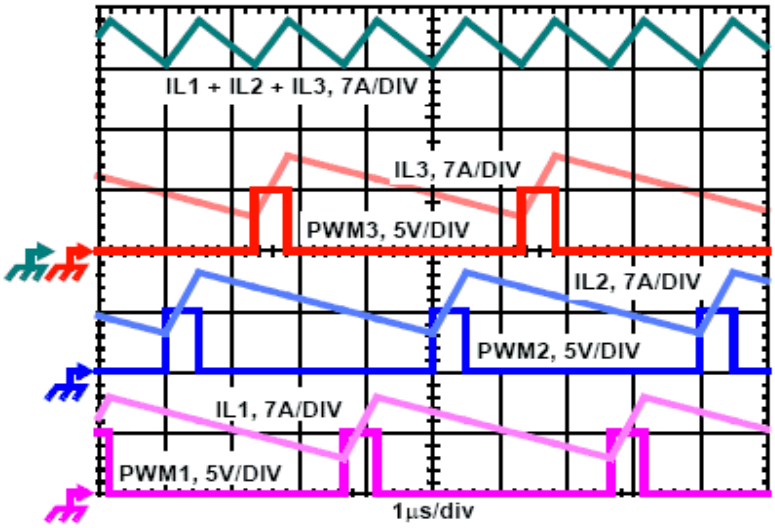
First off, we need to determine both the source and the destination for the signal. In the case of the DDR-II bus, we'll take the driver within the chipset to be the source of the signal, while the bus recovery inside the DRAM device mounted on the DIMM's PCB will be the destination. Now that we have our source and destination, let's take a look at the actual interconnect and power plane. As the interconnect breaks away from the chipset, it will travel a pre-defined distance, called "tuning" which I'll be covering in a minute, to a given pin on the DIMM socket, and up the DIMM's PCB to the DDR-II device. Underneath this data interconnect, there exists a 1.80V power plane.

From a designer's standpoint, they could choose to do something called "flooding," basically creating a large plane of copper to power the DDR bus and for the return currents to flow along. From a power delivery point of view, this would be great because the designer would have a large plate of copper to use to deliver power to the chipset power pins, and the DIMM's power pins. One small problem that arises here is that in reality, it isn't this simple. If you were to take a look at the various layers of a DIMM's PCB, you'd see that unlike our motherboard, the signaling layer is actually placed above a ground layer. If the motherboard designer chooses to route the interconnect traces over the 1.80V power plane, called "power referencing", then it will have problems when it hits the DIMM PCB which is referenced to ground. The return current won't simply stop when this mismatch happens, nor will it simply "jump" from one layer to the other. The return current will in effect find a way to change from the 1.80V power layer on the motherboard to the ground layer of the DIMM. Because this has to happen, the signal will experience some degradation issues. If there are multiple signals taking part in this process at once, you can run into some very serious problems.

In order to avoid such problems, the motherboard designer has to rethink the way they want to lay out these items. A better layout would be to run the interconnect over a second layer ground plane instead of the 1.80V power plane. The designers can embed a ground plane under the DDR-II bus just as easily as a power plane, and that would give the return current a uniform path to travel when returning from the destination to the source. Using the ground plane as the second layer instead of power makes things a little more interesting when it comes to power delivery which is why you'll be able to see some interesting "shapes" on the top and bottom layers that are used for the power delivery.

..:: Design Simulations - Tuning ::..





Yet another important topic that must be addressed by designers is the tuning of the high speed interconnect traces, such as the front side bus connections running from the CPU socket pins to the MCH pins. If you take a look at either the top or bottom layer of a motherboard, you'll notice that

many of the traces do not follow a straight line path, rather they seem to snake around. Some traces do go in a straight line, while a trace next to it might twist and turn. This is referred to as "serpentining" for obvious reasons.

Why is serpentining used? This process is used in order to meet the target lengths for traces that are determined during the simulation process. This is also used in order to make sure that when the signals travel down the traces, they will meet up at the device in unison. Each signal takes a finite amount of time to travel between the source and destination, called the propagation delay, and therefore, varying trace lengths are needed if one signal needs to arrive at the destination at the same time as another.

Since they deal with modern high speed busses that are ever increasing in transfer rates, the designers use something called source synchronous signaling for the clock. This means that the "driver" of the trace also sends clocks with the data signals; these clocks are typically called "strobes". This is another process used in order to properly time the signals so that they reach their end destination when needed so as to not force further delays. This does not mean, however, that all signals must arrive at the destination at the same time. There may actually be a need for a signal to be offset. The designer may also have to deal with other propagation delays of something called differential strobes, and non-differential strobes. Once again, these topics are well beyond the scope of this article.

Once all of the necessary simulations have been run, and all of the manufacturing parameters have been given to them, the designers then run numerous routing studies for the various components that need to be placed on the PCB using all of the tricks that we have just covered. This task it undertaken by the CAD designers who create the layout and make sure that the motherboard can have the needed devices without experiencing any routing problems. The major goal of this part of the process is to see how closely the various components on the motherboard can be placed together. The tuning requirements that are given by the simulations dictate the placement of the traces, as well as attempting to make several areas of the board as compact as possible to leave room for additional onboard features such as audio.

When circumstances arise that a trace may need to transverse through multiple layers of the PCB, we come across something called a "via." These are used in order to pass the trace from one layer to another when needed. When you're talking about single or double layer PCB's, these vias will actually penetrate the entire board. If you have a sound card, or something or the sort lying around, simply hold it up to a light and you might be able to spot a few of these vias that pass through the entire board. When you're working with multiple layer PCB's however, you may not want the vias to completely pass through the PCB, rather only go a certain amount of layers in. These vias are termed "buried" or "blind" vias. You can easily see several blind vias on the surfaces of the motherboard as these vias pass from the surface layer inward. A buried via cannot be seen as it is used to connect the inner layers of the PCB

..:: Processor Power Delivery - Component Background ::..

The topic of processor power delivery is one that brings about several misconceptions, and misunderstandings. Before I delve into the control aspects of the power delivery scheme for the processor, I'm going to quickly go over a few points to explain what the various components are, and what they are responsible for when it comes to the power delivery for the processor. First off, if you take a look at the image above, you'll notice the line of small capacitors along the edge of the heatsink retention mechanism. There are also two light green "hoops" with a coil of wire wrapped around them, these being inductors. When these two electrical components are placed in a circuit and current travels though, you create an oscillator.

Several of you may, or may not be familiar with what an oscillator is, rather only have heard the term used and have an estimated understanding. In order for something to oscillate, the energy the system uses must be converted back and forth between two forms. Let's take a weight attached to a spring hanging from some surface as an example. When you attach the weight to the spring and let it settle, the spring will extend down to a point where it reaches equilibrium. In this instance, it is where both types of energy are themselves equal to zero. If the weight is stretched lower from this point and let go, the system will no longer be in equilibrium and will begin to oscillate. When the weight is stretched, potential energy is added to the system, yet the kinetic energy remains zero because the weight is not yet in motion. When the weight is released, the energy of the system changes from completely potential to completely kinetic. Thus, we have created an oscillation. This same process must apply to a system of capacitors and inductors as well in order for it to create an oscillation. But how?

The two forms of energy involved in the example above were kinetic and potential. In a situation involving a capacitor and inductor, the energy of the system is stored in an electrostatic field and magnetic field. Capacitors store energy in an electrostatic field, while inductors store energy in a magnetic field. As you can now probably understand, in order for a system of a capacitor and inductor to create an oscillation, something has to happen in order to switch back and forth between the two forms of energy. To explain this, let's first take a look at exactly how each of these devices themselves works, and how they store the energy needed for operation.

The way that a capacitor works and stores energy is rather simple. You see, inside the capacitor there are two metal plates called conductors that are separated by a dielectric. This dielectric can be anything from air to something along the lines of a ceramic substance. I won't get too into it, but depending on the dielectric used to separate the two plates of a capacitor, the capacitance can vary greatly depending on the dielectric constant. When a capacitor is placed in a circuit, and charge is allowed to flow, as the current flows through the capacitor, the two plates within attain equal but opposite charges. Thus, an electrostatic field has been formed, and energy stored. The process is a little more involved of course, but you only need a basic understanding of the process.

The way that an inductor stores energy is just as involved as a capacitor, so I'll only be giving you a cursory explanation for those who aren't math or physics buffs. To put it plainly, an inductor is nothing more than a copper wire that is coiled around some material called the "core." As with capacitors, the "core" material can greatly effect the overall inductance. The type of inductors that are readily seen on motherboards are what are referred to as "torodial" inductors. For those who aren't up on their shapes terminology, this name comes about because the core of the inductor is in the general shape of a torus. The other important aspects that determine the capacity of the inductor are the length of the coil, the number of coils / how tightly packed the coils are, and the cross-sectional area of the coil. As a current begins to flow into this coil, a magnetic field begins to

build up and temporarily the coil actually restricts current flow. Once the magnetic field has been completely built, current is then able to flow normally through the coil. When an inductor is placed in a circuit, and the circuit opened after it has been charged, the device in the circuit that is using the electrical energy will continue to work as the magnetic field slowly collapses and gives off the energy that has been stored.

Now that we have a generalized idea as to how each of these devices stores energy, let's examine how they act when placed in a closed circuit together. Let's say for example, that we have a capacitor that has been fully charged and we initially place it into the circuit. As soon as this happens, the capacitor will begin to discharge and the current will begin to flow around the circuit. As the current moves through the circuit, it will then come to the inductor where the magnetic field will begin to form. When the capacitor has been fully discharged, and the magnetic field built, the field will then immediately begin to collapse as the inductor will try to keep current flowing through the circuit. As the field collapses, the energy will then begin to recharge the capacitor and the process will repeat itself until the system runs out of energy due to resistance, unless of course periodically energy is added to the circuit to keep it at a given level. Thus, we have created an oscillator, an imperative factor in the power delivery for the process

..:: Basic Core Voltage Regulation Components ::..

There are two major components that take part in the core voltage power delivery schemes that exist on modern motherboards. The main chip involved would be the multi-phase buck converter chip / PWM controller. An example of such a chip would be the Intersil ISL6556B. Buck controllers such as this are able to drive multiple synchronous-rectified buck-converter channels in parallel. The advantages of a multi-phase power delivery architecture lies in the fact that they utilize interleaved timing in order to multiply the ripple frequency, and lower the input and output ripple currents. By lowering ripple currents, the developer can improve power throughput and efficiency, reduce heat dissipation, and utilize a smaller area on the board for implementation.

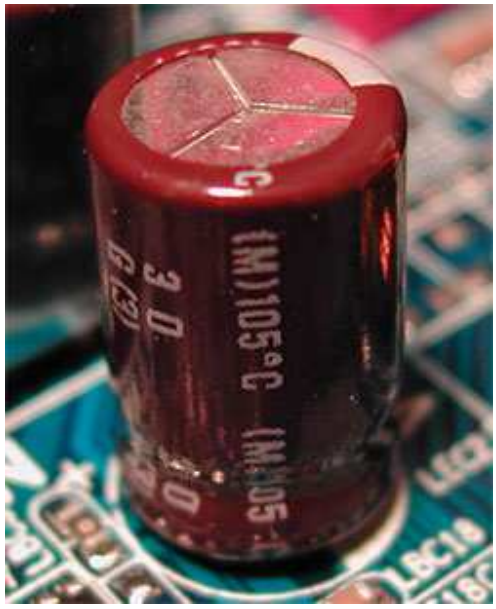
The second component that is involved in the core voltage regulation schemes for modern motherboards is the MOSFET drivers which, via signaling from the PWM controller, switch on and off the MOSFET's it controls, whether that be two or four. An example of a MOSFET driver that can be paired with the ISL6556BCB in two or four channel schemes is the HIP6602B series. When combined with a series of MOSFET's and a PWM controller, you form a complete core voltage regulation scheme for modern processors such as the Intel Pentium 4.

..:: C.V.R. - How Does It Work? ::..

The main control module involved in the core voltage regulation process is the PWM, or Pulse Width Modulation, controller. This chip contains internal and external circuitry that is utilized in order to determine the frequency at which the PWM will cycle, and therefore directly controls the timing when a signal will be sent to each of the MOSFET drivers to switch on the MOSFET. If you click the image above, you'll be able to look at a top level logic diagram of the ISL6556BCB chip.

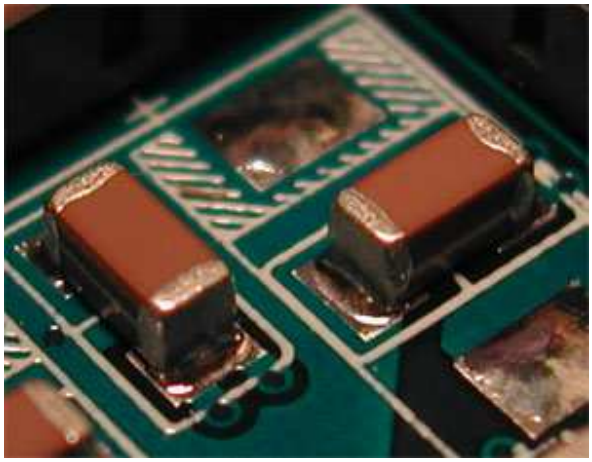
The timing for each of the converter “legs” is determined by however many channels the design will be utilizing, whether it is two, three, or four.

A single cycle for this operation is defined by the time between PWM pulse termination signals. The cycle time is the inverse of the switching frequency which is set by a resistor. A cycle will begin when the clock signals one of the PWM channels to go to low. This will signal the MOSFET driver to turn off the upper channel MOSFET and turn on the Synchronous MOSFET. In a four channel setup, cycles begins 1/4 of a cycle after the previous cycle began, or 1/3 for a three channel setup, etc.



When the PWM signal transitions to low, it will be held there for a pre-determined amount of time in order to get an accurate current sample. Once completed the output PWM is enabled. Here, the voltage is modified and sent to the sawtooth generator where the PWM then transitions to high. When this happens, the MOSFET driver will switch off the Synchronous MOSFET and turn on the upper MOSFET. The current that is sent out by the MOSFET’s then proceeds through the inductors, and on to several output capacitors. The current will then move on to the microprocessor at the accurate voltage and amperage. This process repeats to form the power delivery scheme. The image above shows the waveform example of three interleaved phases from the ISL6566BCB data sheet. There are also two other core components that deal with the core voltage supply, capacitors and inductors.

..:: Capacitors - Common Implementations ::..



The first component that we'll be discussing is that of the capacitor. The most common types of capacitors that can be found on modern motherboards are aluminum electrolytic capacitors, and chip capacitors. Aluminum electrolytic capacitors consist of capacitor paper, electrolyte, cathode aluminum foil, and a layering of aluminum oxide. The aluminum oxide layer serves as the dielectric material for the capacitor as it forms on the anode side of the foil surface. An important aspect of aluminum electrolytic capacitors is that the thickness of the dielectric can be controlled by varying the potential, or voltage, applied. As the potential increases, the amount of aluminum oxide that is deposited onto the anode side of the foil also increases. There are advantages to using such capacitors in computer applications in that several manufacturing techniques allow for high capacitances, while keeping the overall size of the capacitor down.

The second major type of capacitor that can be found on motherboards is what is called a chip capacitor. This type of capacitor is also known as a surface mount capacitor. These capacitors follow along with the same idea as the aluminum electrolytic capacitors, but are far smaller in size, and generally utilize a ceramic as their dielectric. Chip capacitors typically come in three main designs, those being single layer, multilayer, and finally film. Film capacitors are manufactured mainly from plastics such as polystyrene or polyester. The remaining two types, single layer and multilayer, are self explanatory. Multilayer features multiple layers of dielectric, while single layer only features one. Chip capacitors have the advantage of being very small in size, and have excellent thermal characteristics in comparison to other capacitors.

Visually speaking, a chip capacitor looks like a small rectangle, with the interior portion typically being of a clay color. This clay-colored portion is the body of the capacitor which contains the electrodes and dielectric. On the ends, we see exposed metal which serve as the termination points. The materials that make up these termination points are typically silver, with a coating of nickel to serve as a barrier, and finally tin plating. A common area where you'll notice chip capacitors is within the central area of the processor socket, or around the socket where the heatsink would be mounted. They can also be readily seen in use on graphics adapters, sound cards, etc. as they are quite common.