Chapter 2
Landscape: History, Present Barriers, and The Road Forward

The more I learn, the more I realize I don't know. The more I realize I don't know, the more I want to learn.

Albert Einstein

Abstract Software Thermal Management is a systems level concern that considers schematic capture, Printed Circuit Board (PCB) layout, mechanical design, materials science, software engineering, and use-case scenarios. The landscape is large and approaches are not standardized. This chapter reviews the history of Moore's Law, the limitations of parallelism, and the special role that software engineers have to play when managing thermals in an embedded system.

2.1 History

The number of embedded devices in the world is growing, and that growth will accelerate in the coming decades. In a 2011 report by Cisco IBSG, it is predicted that by 2015, we will have 25 billion connected devices, and by 2020 there will be 50 billion connected devices in the world. Figure 2.1 depicts this expected trend. Whether or not the growth rate will be slower or faster than this is up for debate. However, what is certain is that the number of electronic devices will be increasing significantly in the coming decades.

As the number of devices increase, so inevitably will the volume of microprocessor sales, and these microprocessors are becoming more and more capable of complex computation. In the 2013 McClean Report: A Complete Analysis and Forecast of the Integrated Circuit Industry, it is shown that the trend in microprocessor sales is moving from 4- and 8-bit microprocessors to 16- and 32-bit processors. More powerful processors also consume more power and consequently produce more heat. This trend is shown in Fig. 2.2.

M. Benson, *The Art of Software Thermal Management for Embedded Systems*,
DOI: 10.1007/978-1-4939-0298-9_2,

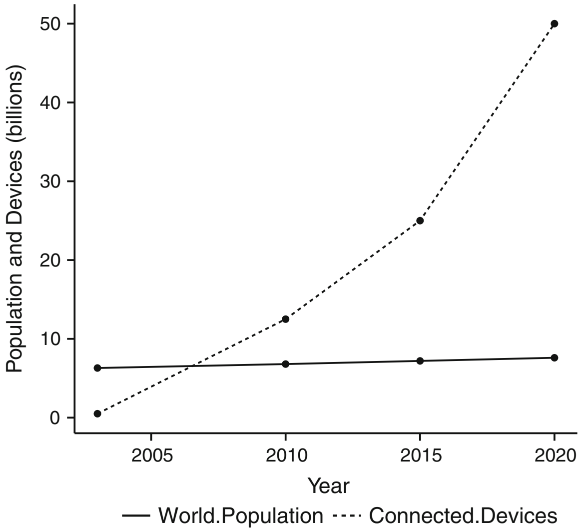

Fig. 2.1 Internet of Things (IoT) growth forecast according to a Cisco IBSG report titled *The Internet of Things: How the Next Evolution of the Internet Is Changing Everything*, 2011. Note that in 2007–2008, the number of connected devices in the world exceeded the number of people in the world (world population). This trend will inevitably continue to grow as more non-consumer devices become connected such as industrial equipment, public utilities, transportation and traffic control systems, and so on

In a January 2013 report by Global Information Inc. and BCC Research, the market for thermal management technologies (fans, blowers, heat sinks, materials, and substrates) was valued at $6.7 billion in 2011, reached $7 billion in 2012, and is expected to reach $10.1 billion in 2017 after increasing at an expected five-year Compound Annual Growth Rate (CAGR) of 7.6 %. Most of this growth is due to the increased need for these technologies in the computing industry due to higher computational needs, as shown in Fig. 2.3.

This trend shows clearly that there is an increasing need in computing for ways to transfer heat. Since heat is a byproduct of power consumption, we must understand and control power in order to control heat.

Similar to a physician that diagnoses a medical condition based on a set of observed symptoms, with heat problems in embedded systems a majority of existing solutions focus on transferring heat away (treating the symptom), yet the root cause comes down to power dissipation (the cause).

At its core, computing requires power. Power in turn requires electricity which produces heat. Heat, when present in large enough and focused enough quantities can cause thermal symptoms such as reduced component reliability, reduced performance, increased safety risks, and total system failure. The diagram in Fig. 2.4 shows the relationship between computing requirements, power, heat, and thermal symptoms that result.

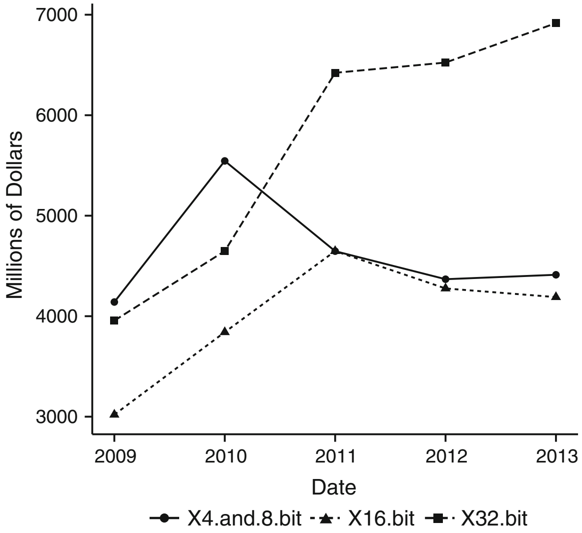

Fig. 2.2 Microcontroller Unit (MCU) market history and sales forecast, from *The 2013 McClean Report: A Complete Analysis and Forecast of the Integrated Circuit Industry*. As computational demands continue to grow, there is showing to be a clear trend towards 32-bit processors. 32-bit processors are more capable of computation (math, video, audio, etc.), but also produces more heat. This trend towards 32-bit processors makes it even more important that dynamic and static thermal performance is managed for these processors

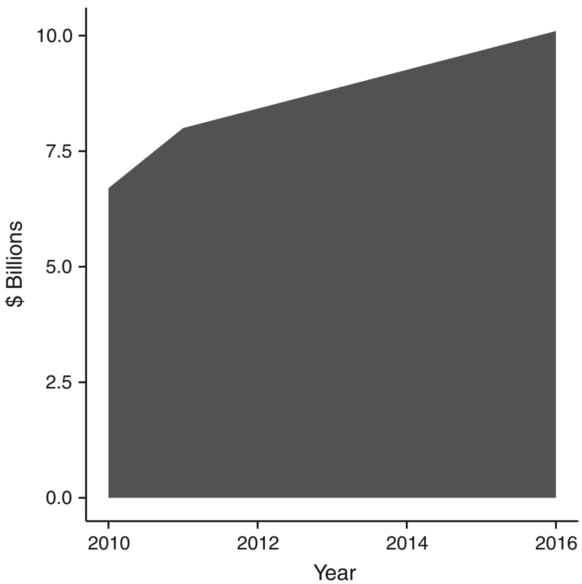

Fig. 2.3 World thermal management market trends, 2010–2016 from *The Market for Thermal Management Technologies* report by Global Information Inc. and BCC Research, January 2013. Thermal management technologies considered include fans, blowers, heat sinks, materials, and substrates. Data for 2014–2016 is predicted

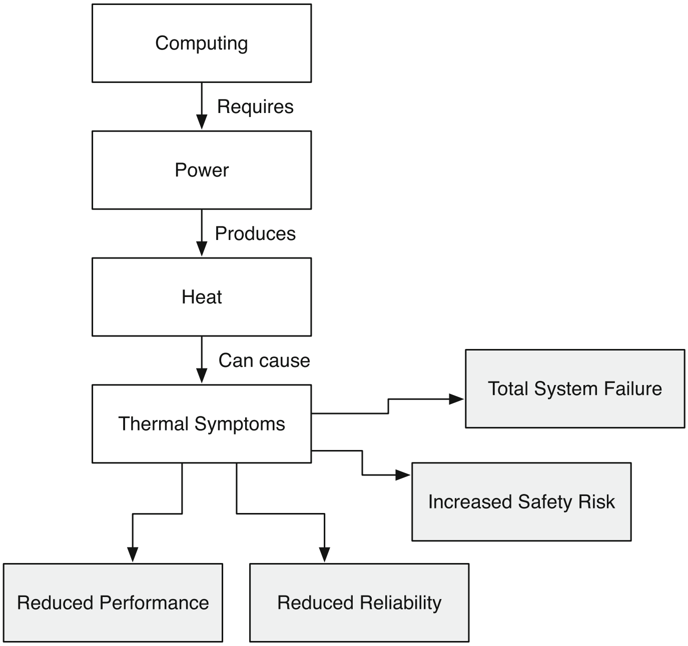

Fig. 2.4 The relationship between computing, power, heat, and thermal symptoms in embedded systems is shown here. Computational demands require power to operate. Power, when consumed, dissipates and produces heat as a byproduct. Heat, when presented in high quantities and in focused areas can cause negative thermal symptoms to occur. Examples of thermal symptoms include reduced component performance, reduced battery life (batteries discharge at a faster rate when temperatures are higher), reduced reliability (components fail if they exceed their Recommended Operating Conditions), increased safety risk (as is the case when an integrated circuit fails due to overheating and the failure modes are unpredictable), and total system failure if the processor or key components around it are compromised or cease to work

If an integrated circuit operates within the specified temperature range by the manufacturer of the product, negative thermal symptoms do not usually occur. However, when operating in environments with extremely high ambient temperatures, or when microprocessors must operate at a high level of computation (and power consumption) for a long period of time to satisfy required demands, it can become a serious problem and lead to system reliability and safety issues.

Integrated circuits are typically manufactured and sold under certain grades, which have different operating temperature ranges. Common integrated circuit temperature ranges are shown in Table 2.1.

The commercial and industrial grades are relatively easy to find when selecting electronic components for a design. However, the military grade versions of parts are only used in special scenarios on special projects, and since the volume of demand is lower it means that military-grade parts are harder to find, or perhaps impossible if the vendor has not chosen to create a military-grade version of their part.

Table 2.1 Common integrated circuit temperature ranges

Grade	Min	Max
Commercial	0 °C	70 °C
Industrial	−40 °C	85 °C
Military	−55 °C	125 °C

Thermal fatigue is particularly insidious as peak heat events can cause fatigue and a reduced lifetime for an electrical component without displaying external symptoms but have an increased chance of failure at a later point in time. For safety-critical equipment, it is simply not an option to wait and see—we must be proactive. By picking industrial-grade parts, we can widen the Recommended Operating Conditions and meet more difficult environmental conditions and requirements. Industrial- and military-grade parts usually cost more than their commercial counterparts, and may carry different lead times (the duration of time between when the parts are ordered and when they are expected to be delivered).

Summary

- The number of embedded devices in the world is growing at a high rate.
- The industry is moving to 32-bit processors as those processors are more capable of meeting complex computational demands, while still being capable of being put in very low-power suspend modes.
- The world thermal management market (fans, blowers, heat sinks, materials, and substrates) is growing, indicating that there is an increasing need for mechanical components that conduct heat or provide more efficient means for convection to occur, thereby transferring heat more effectively in electronic systems.
- Heat is a natural byproduct of power in electronic systems. If heat is produced in large enough quantities and in concentrated regions, thermal fatigue or failure may occur if the Recommended Operating Condition for the part is violated.
- Electronic components are manufactured in different grades (commercial, industrial, military are most common) to help meet needs of the application by guaranteeing that the part will operate normally across the entire temperature range.

2.2 Barriers

The story of SOFTWARE THERMAL MANAGEMENT, however, is not that simple. The thermal performance of a dynamic and complex embedded system is difficult to understand, difficult to model, and difficult to manage. At each step in the design process, there are hurdles to overcome:

- Defining requirements for the system is a challenge in and of itself. Understanding what users want and need, and in what context they are in when they need to do their work effectively using the product is a field of its own [1–11].
- Defining the operating environment can be tricky. Many academic papers seek to model and describe the way in which we should characterize and define the end operating environment for an electronic device [12–15].
- Designing electronics, including the schematic capture and also the Printed Circuit Board (PCB) layout, is where many of the key decisions are made that ultimately affect the ability for the system to transfer heat, and for the software to be able to control the power and thermal performance. For instance, where components are placed on the board, how those components come into contact with the PCB or with the case make a difference. Where vias are placed makes a difference.[1] The way in which the circuits are designed such that they may or may not be turned off by software when not in use has a major influence on the thermal performance of a system, and those decisions are made at design time. Numerous resources exist on this topic [16–27].
- Picking a processor that meets the functional needs of an electronics system, and also meets the thermal constraints that the system must operate in, can be difficult to do well. In many cases, multiple processors may meet the needs of the system requirements, but be from different vendors, with different software development kits (SDKs) that have varying levels of maturity and utility, and may have subtly-different capabilities when it comes to managing power and heat. Since understanding true requirements of the end system for power and heat is difficult picking a processor can be problematic.
- Using all the features of a processor that are available to you is important, but takes time, and that time is difficult to estimate. Especially with complex system-on-chip (SoC) processors that have heterogeneous cores that are capable of advanced multimedia processing, and also very low power, have many many knobs to turn and levers to pull that can be used together to achieve very low dynamic and static power. However, the data sheets that accompany these processors are large and complex, and sorting through it all can be a barrier to the design and implementation of optimal thermal performance for an embedded system.
- Coordinating resources (peripherals) in an embedded system to work in concert with each other to meet the thermal performance requirements can be tricky. Chapter 5 addresses this.
- Because thermal fatigue can be almost impossible to empirically observe, the true damage of high thermal peaks on an integrated circuit can be confounding. It's easier when a processor completely overheats since discoloration of the electronics will be visually apparent, and the system will cease to operate all together.

These and other challenges are what we are up against. There are some known techniques to deal with thermal problems in embedded computing systems, some frameworks that are beginning to emerge such as in the Linux kernel, but it is still in

[1] A via (Latin for path or way) is an electrical connection between layers in a physical electronic circuit (e.g. a PCB) that goes through the plane of one or more adjacent layers.

many ways a black art that can only be mastered by gaining a wealth of experience through a combination of theory and pragmatism. The following sections describe these barriers in more detail.

Summary

- Planning for and working through thermal performance issues in embedded systems can be viewed as a sequence of barriers. Our task is to overcome those barriers and also lower them for subsequent designs.
- There are known techniques and emerging frameworks for dealing with thermal problems, but the introduction of software engineers to the complexities of thermodynamics and electronics component design is relatively new.

2.2.1 Moore's Limitations

Moore's Law is the observation made in 1965 by Gordon Moore, co-founder of Intel, that the number of transistors per square inch on integrated circuits had doubled every year since the integrated circuit was invented. In his landmark paper, Moore predicted that this trend would continue for the foreseeable future [28].

In subsequent years, the pace has slowed, but data density has continued to double approximately every 18 months. Hence, 18 months is the current accepted definition of Moore's Law.

To illustrate Moore's Law, see Fig. 2.5 which shows Intel transistor density for major new processors from 1970 to 2010.

Summary

- Moore's Law says that the number of transistors per square inch has and will continue to increase at an exponential rate.
- Moore's Law has remained amazingly accurate for nearly 50 years (although we'll see some caveats to this in Sect. 2.2.2.

2.2.2 Thermal Wall

Interestingly, if we look closer, an odd phenomenon has emerged: although transistor density has continued to climb, CPU speed has not followed in the same manner. Clock speeds and power consumption have both leveled off starting in 2005 as

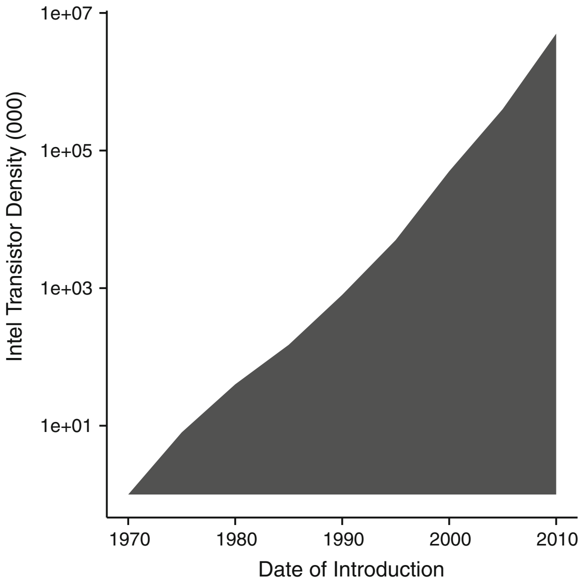

Fig. 2.5 Moore's Law, as it applies to Intel CPU transistor density. This graph shows the CPU transistor counts against dates of introduction over time. Note the logarithmic vertical scale of this graph indicating continued exponential growth

shown in Figs. 2.6 and 2.8. What's happening here? The answer has to do with thermodynamics and the Law of Dynamic Power.

We are reaching physical limits of what we can do in terms of heat transfer. As processor transistor density has increased, so has the amount of power that is being consumed. As that power is consumed, heat is generated, and our limited ability to transfer that heat to the surrounding ambient air mass is prohibiting us from continuing to make faster and faster processors.

Summary

- We have reached a thermal wall in that CPU speeds are not able to exceed 3 GHz without active cooling.
- This trend is caused by the fact that at high frequencies, so much heat is generated as a by-product of power dissipation fundamentals, that we are unable to transfer that heat away from the processor fast enough.
- The answer is to reduce the frequency of a processor, or add more cores (each at a lower frequency) to achieve similar or higher computational performance, but at a lower level of power consumption, and thereby much better thermal performance.

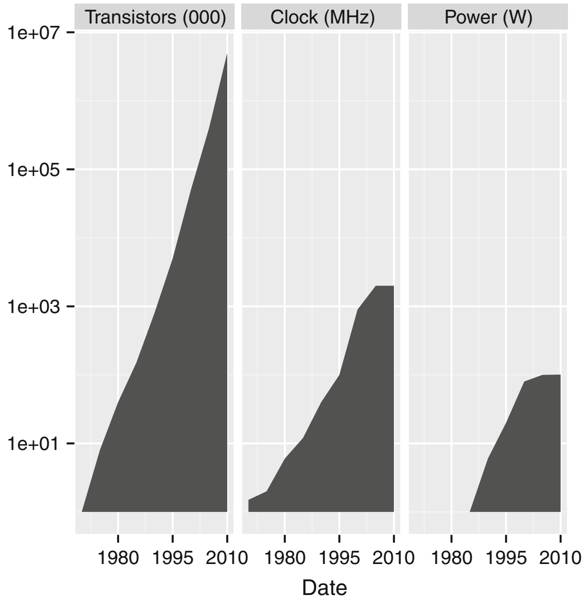

Fig. 2.6 Transistor, clock, and power facets for Intel CPUs over time. In this graph, you can see that the transistor density has continued climbing according to Moore's Law. However, power consumption and CPU frequency have leveled off as we have bumped up against signal integrity issues and issues transferring heat from a device with such a small thermal mass to ambient

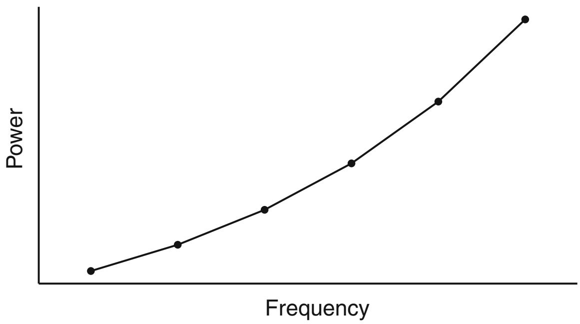

Fig. 2.7 The Law of Dynamic Power, when graphed as a function of processor frequency, shows that as the frequency of a processor increases, the amount of power that is required to charge and discharge the processor's capacitive load grows at an exponential rate. Because our goal with SOFTWARE THERMAL MANAGEMENT is to manage the thermal performance of a system, we should be looking to reduce the processor frequency and voltage as often as possible and for as long as possible to ensure that unnecessary processing power and unnecessary input voltage levels are minimized

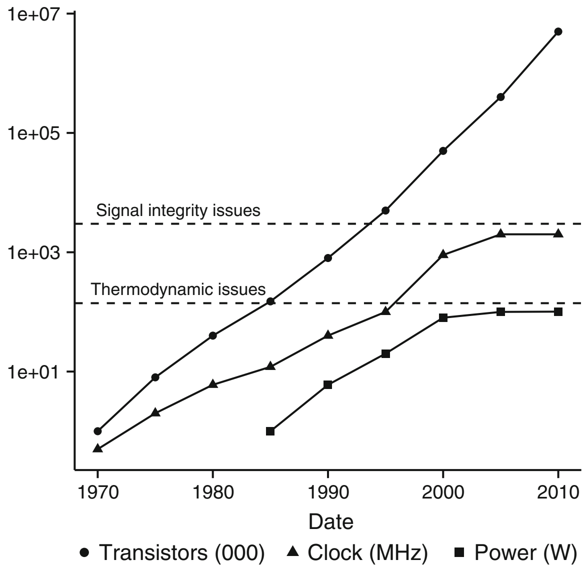

Fig. 2.8 As processors have grown to be more powerful, they have also consumed more power and produced more heat to the point where the industry has now moved to homogeneous or heterogeneous multicore solutions for processor-intensive applications so that each core can run at a lower frequency and the processor will thus consume significantly less power as a result [29]. Dynamic power in CMOS integrated circuits is a nonlinear function of capacitance, frequency, and voltage squared (Eq. 2.1, Fig. 2.7). The nonlinear nature of this relationship is especially important because it tells us that as the switching frequency gets higher, the amount of power (and consequently, heat) grows exponentially

2.2.3 Dynamic Power

At the base, the reason why clock speeds have not kept up with Moore's Law is because of the Law of Dynamic Power. The Law of Dynamic Power is modeled by Eq. 2.1 which characterizes the power lost as the processor charges and discharges its capacitive loads. As you can see from the V^2 portion of the equation, reducing the voltage has the largest impact on total power consumption.

$$P = CV^2 f \tag{2.1}$$

The Law of Dynamic Power is shown in Eq. 2.1, where P is power, C is capacitance, V is volts, and f is switching frequency. As the speed of a CPU increases (switching frequency), the amount of power required to motivate those flops also increases. This, in turn, increases the amount of heat that is generated, which becomes more and more difficult to remove given the relatively small thermal mass of a processor.[2]

The fact that the dynamic power equation is a nonlinear equation is especially interesting. From it, we can draw a conclusion that for low-power embedded systems, we should be looking to reduce voltage where possible, necessarily also reducing the frequency, which in turns reduces power, and thereby also reduces heat.

As we'll see in the remainder of this book, much of the theory, techniques, frameworks, and practical applications of SOFTWARE THERMAL MANAGEMENT hinges on this Law of Dynamic Power. If we are clever, we can maneuver the curve traversing up and down the curve freely, tune the curve to our advantage using Adaptive Voltage Scaling (AVS) technologies, spend more time in the optimal (lower) parts of the curve, and scale the curve quickly and on demand with select fast-boot optimizations.

To demonstrate the relationship between power and frequency, let's look at the Intel Pentium M processor family. The Intel Pentium M processor family was a set of 32-bit single-core x86 microprocessors introduced in March 2003. The "M" stands for mobile and the processors featured a series of operating points which could be selected. Each operating point was a combination of {frequency, voltage} pairs defining a set of states that the processor could run in. The six operating points for Intel Pentium M processors are shown in Fig. 2.9.

There is a class of System-on-Chip (SoC) processors such as the TI OMAP, Qualcomm Snapdragon, and Nvidia Tegra line of processors that combine multiple ARM processing cores with a graphics engine and a Digital Signal Processor (DSP) all into one package. This type of SoC is often used in cell phones and is capable of massive computational power when needed, but can scale down to very low power modes when idle. From a thermal perspective, if these chips are run at full speed for a long period of time, they will breach their recommended operating temperature range, and could damage themselves and/or the surrounding components.

Summary

- Dynamic power is a combination of capacitance, frequency, and voltage squared.
- The relationship modeled by the Law of Dynamic Power shows that a major component of the power consumption of an integrated circuit is the input voltage required to scale up the frequency.
- To reduce dynamic power, the best way to do so is to reduce voltage. Reducing the voltage also requires that the frequency is reduced as well.

---

[2] For a given processor, C is fixed. However, V and f vary. Caveats: (1) Some CPU instructions use less energy per tick of the CPU clock than others; (2) Static power consumption of the CPU (power consumed when the CPU is not doing meaningful work) is not represented by this equation. Static power consumption *does* vary with temperature, however. Warm electrons, especially those exposed to a stronger electromagnetic field are more likely to migrate across gates, and are considered "gate leakage" current, adding to the total static power consumption of the CPU.

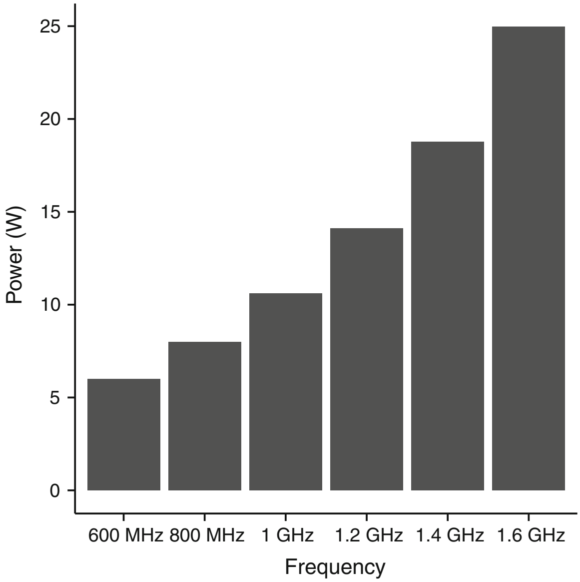

Fig. 2.9 Power versus frequency for the Intel Pentium M processor family running at 1.6 GHz for its six frequency/voltage operating points (not to scale). What this graph shows is that frequency and power are related, and that relationship is not linear. As frequency grows, thanks to the V^2 part of the Law of Dynamic Power (Eq. 2.1), power also grows at an exponential rate

- New SoC chips are built to scale up and down based on current computational demand. These processors are especially useful in cellphones and tablets.
- In the field of SOFTWARE THERMAL MANAGEMENT, our job is to navigate and manipulate the dynamic power curve in our favor—the remaining chapters and sections of this book dive into those concepts in more detail.

2.2.4 Multicore Promise

The trend towards multicore starting in 2005 was caused by the thermal wall, where we have run up against the limits of CPU speed (signal integrity issues) and power dissipation (heat transfer issues) for ever-shrinking thermal masses.

Moving to multiple smaller cores instead of one big one has a number of advantages from a thermal perspective. Look back at the Law of Dynamic Power (Eq. 2.1). Because of the quadratic relationship between frequency and voltage, running two cores at half the speed consumes less power than one core running at full speed as shown in Fig. 2.10.

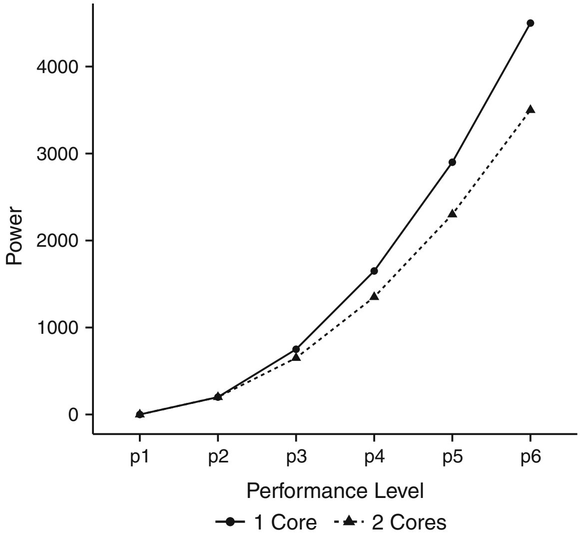

Fig. 2.10 A single core running at 2x the frequency will always consume more power than its multicore equivalent. This fact is based on the Law of Dynamic Power, and should drive us towards two ends: (1) reduce power whenever possible, and (2) if we have the option of running multiple cores instead of one, we will benefit by having more processing power per unit power consumed. This benefit comes at the added complexity of software parallelism techniques that are often error-prone

With the move to multicore, we gain improvements in speed and computational ability while consuming less power and producing less heat in exchange for higher parallelism complexity in software. Said another way, it doesn't do us any good to have multiple cores if we are only running a single-threaded program.

When discussing the concept of multicore, it's important to point out that there are two broad types of multicore solutions: Symmetric Multiprocessing (SMP) and Asymmetric Multiprocessing (AMP).

- Symmetric Multiprocessing consists of N homogeneous cores (i.e. two ARM cores). Typically one operating system and one software system manages all of the cores.
- Asymmetric Multiprocessing, on the other hand, consists of N heterogeneous core (e.g. ARM + digital signal processor + image signal processor + graphics engine). In this case, the cores are often managed by multiple operating systems and software sub-systems.

In the remainder of this book, when we talk about multicore, we'll most often be talking about hybrid multicore solution (symmetric + asymmetric) which may contain multiple homogeneous ARM cores, and also heterogeneous processing cores such as an ISP and DSP all in the same package.

Regarding thermal performance, multicore solutions offer us the following benefits:

- More performance
- Less power consumed
- Less heat produced

Summary

- The trend towards multicore solutions in 2005 was caused by the troubles that were caused by high processor frequencies requiring higher input voltage and producing too much heat too quickly to be able to transfer it effectively.
- Multicore solutions allow us to have similar or better computational performance at the expense of higher complexity in the software design so as to support parallelism.

2.2.5 Amdahl's Wet Blanket

With multicore solutions, however, the picture is not as rosy as you might think. At first, it may seem best to continue to add add more cores. If adding more cores allows us to have more processing power at an equivalent or better level of power consumption and thermal performance, why not just continue adding more and more cores?

Amdahl's Law was formulated to model the amount of speedup we receive when we add a new core. The model is based on two principles. The first is that by adding more cores, we can achieve higher levels of computation. The second principle is that there are limits to how much we can speed up our computational engine since the amount of speedup (according to Amdahl) is limited by the amount of our software program that can be parallelized.

Amdahl's Law can be simply stated as shown in Eq. 2.2.

$$1/(1 - P) \tag{2.2}$$

Amdahl's Law is stated here, where P is the proportion of a program that can be made parallel. Said another way, a program can only be sped up to be as fast as it's largest single-threaded part. If a program cannot be parallelized, throwing 100 cores at the problem will not speed up our program, nor will it spread out the computation across multiple cores, and hence will also not spread out the heat. Figure 2.11 shows a graphical depiction of Amdahl's Law as the number of cores approaches infinity.

Moving to multiple cores and running with a high degree of parallelism can help our heat problems, but not eliminate them. High processing power, such as with

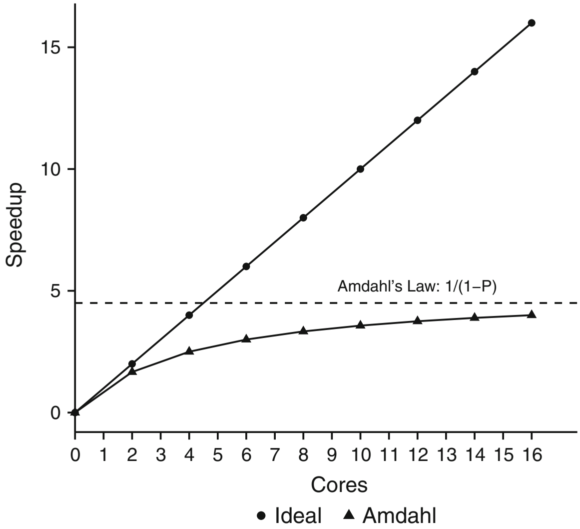

Fig. 2.11 Amdahl's Law constrains the amount of processor speedup we can achieve by adding multiple cores. The limit is based on the proportion of the software program that can be parallelized

portable, battery powered, multimedia-capable devices will cause significant heat problems that we still need to solve. In the next section, we'll talk about these heat problems in depth.

Summary

- Amdahl's Law models the amount of speedup we can expect to achieve when we add more processing cores to a given solution.
- Amdahl's Law says that the amount of speedup is limited by the proportion of the software program that can be parallelized.
- Moving to a multicore solution can help ease our SOFTWARE THERMAL MANAGEMENT problems. However, multicore is not the end solution. In order to really do SOFTWARE THERMAL MANAGEMENT well, we must learn to traverse and tame the dynamic power curve.

2.2.6 Temperature Limits

Every microcontroller comes with a Recommended Operating Condition range. If the junction temperature inside the case of the microcontroller exceeds the recommended levels, the chip may not operate correctly, have degraded performance, take on thermal fatigue (shortening its life span), or may cease to operate all together. A simplified graphical view of recommended operating temperature ranges for microcontrollers is shown in Fig. 2.12.

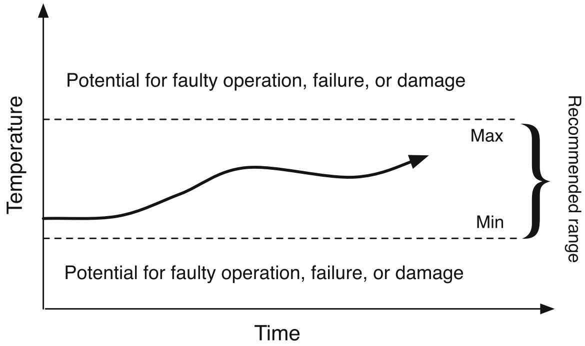

Fig. 2.12 Microcontroller recommended operating temperature ranges specify the upper and lower limits of temperature within which the microcontroller can operate safely. If the junction temperature of the microcontroller (inside its case) exceeds this range specification, fatigue or damage may occur to the part. Thermal fatigue is particularly difficult to notice since visible evidence is often not available, and since the part may continue to operate, the problems lurking within the case may not exhibit themselves until later

Semiconductor parts such as microprocessors are often specified for use in commercial applications, giving a normal operating temperature range of 0–70 °C. Additionally, for industrial applications, there is a need for a wider range, and so those parts are often specified to operate normally in the −40 –85 °C range. See Table 2.1 for details.

There are some applications such as within military, oil and gas, and automotive industries, that require an even wider range. Because of the lower volume demands for these industries, these parts are often harder to acquire, or perhaps not available at all, except by special contract.

These temperature ranges are formalized into standard measures called Recommended Operating Conditions (ROC), and Absolute Maximum Ratings (AMR). ROC specifies the recommended temperature range that is pretty much guaranteed to operate safely. AMR is the temperature range that if exceeded is virtually guaranteed to inflict damage to the part. See Fig. 2.13 for a visual representation of ROC and AMR in relation to one another.

Summary

- Temperature limits for an integrated circuit specify the range that is safe for the integrated circuit to operate. Exceeding these limits will cause thermal fatigue or a shortened life expectancy for the part.
- Recommended Operating Conditions (ROC) specify the safe operating range for the part.
- Absolute Maximum Ratings (AMR) specify the range that, if exceeded, will almost certainly cause the part to fail to operate correctly.

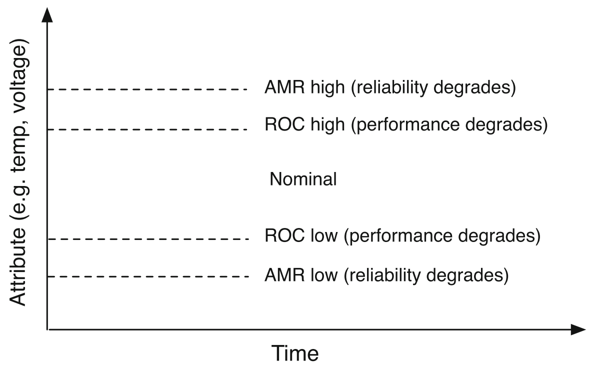

Fig. 2.13 Recommended Operating Conditions (ROC) and Absolute Maximum Ratings (AMR) for electronic components define the temperature ranges within which define safe operation. The ROC defines a safe and reliable operating temperature range. Exceeding the ROC will cause the part to perform sub-optimally, or perhaps exhibit thermal fatigue. Exceeding the AMR will significantly increase the risk of failure for the part

2.2.6.1 Recommended Operating Conditions

Recommended Operating Conditions (ROC) provided by part manufacturers include voltage levels and temperature ranges. By specifying ROCs, manufacturers are not usually guaranteeing the reliability of the part under those conditions. Rather, they are documenting the conditions under which they have conducted their own tests, and can be highly certain the part will operate normally under those conditions.

Parts may fail for many reasons. However, the rate of failure will increase significantly if the part operates outside of its designated ROC.

Summary

- Recommended Operating Conditions (ROC) specify a temperature range for the part that is safe for the part to operate within.
- If the ROC is violated, the part may not operate in an optimal manner, may exhibit symptoms from thermal fatigue over time, or may fail all together.

2.2.6.2 Absolute Maximum Ratings

The Absolute Maximum Rating (AMR) section in a datasheet provides limits on operational and environmental parameters, including power, supply and input voltages, operating temperature, junction temperature, and storage temperature.

Definition 2.1. The International Electrotechnical Commission (IEC) defines Absolute Maximum Ratings as "limiting values of operating and environmental conditions applicable to any electronic device of a specific type as defined by its published data, which should not be exceeded under the worst possible conditions. These values are chosen by the device manufacturer to provide acceptable serviceability of the device, taking no responsibility for equipment variations, and the effects of changes in operating conditions due to variations in the characteristics of the device under consideration and all other electronic devices in the equipment. The equipment manufacturer should design so that, initially and throughout life, no absolute-maximum value for the intended service is exceeded with any device under the worst probable operating conditions with respect to supply voltage variation, equipment component variation, equipment control adjustment, load variations, signal variation, environmental conditions, and variation in characteristics of the device under consideration and of all other electronic devices in the equipment [30]."

Said another way, the manufacturer of the part selects the AMR values, and the Original Equipment Manufacturers (OEMs) that integrate those parts into products and systems are responsible for assuring that the conditions specified by the manufacturer are not exceeded.

Part manufacturers provide AMRs as limits for reliable operation and do not guarantee the electrical performance or operation of the part beyond the AMRs. Exceeding the AMRs will significantly increase the risk of physical damage occurring for the device.

Summary

- The Absolute Maximum Rating (AMR) for an integrated circuit defines the temperature range for the part such that if that range is exceeded, the risk of thermal fatigue or complete failure rises significantly.
- When selecting components for an embedded systems design, ensure that the AMR is appropriate for the environmental operating conditions and thermal use cases defined for the system.

2.2.7 Embedded Complications

Embedded systems (as opposed to PCs or supercomputers) are typically smaller in size and often enclosed or sealed inside cases. The unique nature of embedded systems design, and the categories of use cases that embedded systems designs are made to address, carry a heavy thermal burden for the embedded systems designer. This is especially true when the embedded system must cycle between high computation demand such as with multimedia video or audio use cases, and very low power modes to conserve precious battery power. Here are some specific reasons why heat is problematic in embedded systems:

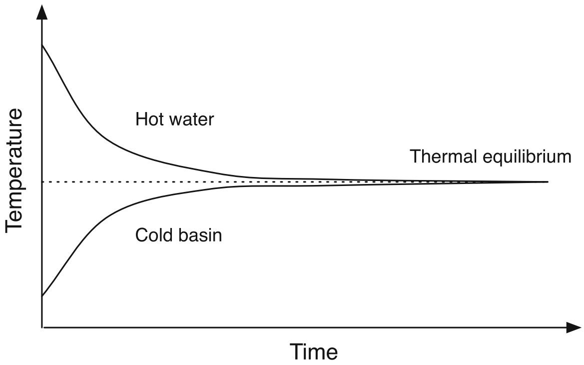

Fig. 2.14 The rate of heat transfer slows over time as thermal equilibrium approaches. In this example, a hot glass of water is placed in a larger basin of cold water. Over time, the heat will transfer from the hot glass to the cold basin. However, the rate of heat transfer will slow as the two temperatures approach each other

1. *Higher ambient temperatures.* Embedded systems often operate in more extreme temperature conditions. In addition, the ambient temperature inside the case of an embedded system can also be higher than non-embedded devices due to sealed enclosures and limited avenues for convection. Because of this, there is less margin between the ambient temperature and the maximum die temperature (AMR) of the part. Since heat transfers more slowly as thermal equilibrium is reached, this can cause problems. For an example, consider a glass of hot water sitting in a larger basin of cold water. Over time, heat transfers from the hot glass to the cold basin. However, the rate of heat transfer slows over time as thermal equilibrium is approached. See Fig. 2.14.
2. *Smaller thermal mass.* This is due to the fact that ICs are very small, and it's difficult to remove heat effectively from elements with small thermal masses. When heat is hard to remove, it becomes easier for high peak temperatures to cause thermal fatigue or failure. See Fig. 2.15.
3. *Sealed enclosures.* For embedded systems that must be water proof, or sealed from environmental elements, the ways in which heat can transfer to ambient are limited. This fact causes additional thermal issues for embedded systems.
4. *Extended usage and long-life support.* The overall life of integrated circuits is decreased when exposed to elevated temperatures beyond the ROC or AMR. Embedded systems that are installed in factory equipment, or are used in industrial or medical settings will often have a very long life (5–15 years). Because of this, thermal fatigue that occurs early in the life of the part will have a greater chance of affecting the reliability of the part since the life time for that part is very long. The fact that embedded systems often have extended usage and long-life support from OEMs causes additional ares of concern for thermal performance and susceptibility to thermal fatigue.

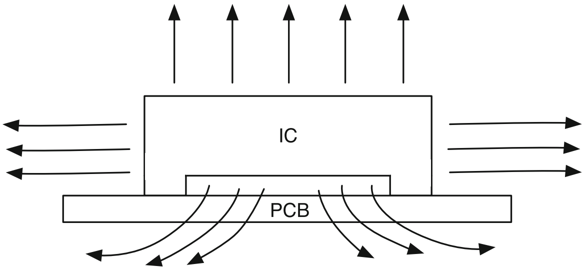

Fig. 2.15 Heat transfers from integrated circuits in all directions. Pads placed under (and optionally, above) the processor, can help conduct the heat more quickly, allowing the heat to escape faster

5. *Exposing batteries to elevated temperatures increases self-discharge.* Although reliability is not typically affected, the performance of the battery, and hence the system, is reduced with elevated temperatures. Battery chemistry is activated when batteries are used and power is consumed. The rate of power discharged from a battery from the chemistry used in batteries is accelerated at higher temperatures. Embedded systems often have batteries, and if so, this fact can subtract from ideal thermal performance.
6. *To help with Radio Frequency (RF) performance, many case designs are made out of glass or plastic.* However, these materials tend to insulate heat rather than conduct efficiently to an outside environment. Aluminum cases have good conductive heat properties that can efficiently transfer heat. However, machining aluminum is costly, and may not make sense unless the unit price or the projected volume of the device is high.

Summary

- Embedded systems are small, usually in sealed enclosures, and made of heat-insulating materials. These factors make the problem of solving for optimal thermal performance especially difficult.
- Thermal problems are exacerbated when the peak computational requirements of a system are high and the expected lifetime of the device is long.

2.3 Solutions

When solving thermal problems, there are three main approaches we can take:

1. *Reduce power consumption.* The best way to generate less heat is to consume less power. Because of the laws of thermodynamics (i.e. conservation of energy), and system behaviors governed by entropy and enthalpy equilibrium, the best way for us to not generate heat in the first place is to consume as little power as possible.

2. *Transfer heat efficiently.* Once heat is generated, the job then becomes to transfer it by providing an efficient path from the device to the environment via thermal pads, epoxy, clips, fans, liquid, or any number of other methods.
3. *Define the environment.* If all else fails, a final way to limit heat is to define the operating environment. For example, specifying that the ambient operating temperature for the device must be at or lower than a given temperature is one way to constrain the environment to your advantage. However, most often the environment defines itself; hence this dimension of thermal performance management can only be used if you have complete control or intimate knowledge of the operating environment.

Of these three categories, software and electronics design is primarily suited to address #1. Heat in an embedded system is a byproduct of power. By focusing on software techniques to reduce power, we can effect the thermal performance of the device and stop heat from being generated at its source. The best way to manage heat is to not consume power.

The following sections take a look at each of these approaches in detail.

Summary

- Solving thermal problems can be done in three ways: by reducing power consumption, by transferring heat efficiently, and by defining the environment.
- SOFTWARE THERMAL MANAGEMENT, together with electronics design, is primarily suited to reduce power consumption as a means to manage heat.

2.3.1 Reduce Power Consumption

As we aim to manage heat in embedded systems, our goal must be first and foremost to not generate heat in the first place. Fortunately (or unfortunately), software engineers have considerable sway in this area.

Going back to the basic principle that heat is a byproduct of energy consumed by electronic systems, the job of the software engineer is to make our embedded systems consume less power and thereby produce less heat.

Power in electronics is composed of two parts: dynamic power and static power. Static power deals with leakage currents and low-level power that is lost during transmission or wasted, even if the integrated circuit or processor is not doing any useful computation. Dynamic power, however has considerable influence on the thermal performance of a system, and determines the peak times of power consumption and thermal output. This book focuses primarily on dynamic power since that is the main contributor to poor thermal performance in embedded systems.

Summary

- One way to manage heat in an embedded system, which software has the most control over, is to reduce power consumption. Since power dissipation produces heat as a byproduct, we can reduce heat by reducing power.
- Power in an embedded system is composed of two parts: dynamic power and static power. Static power, although important for saving battery life and reducing environmental impact, is not as big of a contributor to thermal output, and so this book focuses primarily on dynamic power as it is the prime contributor to peak thermal events.

2.3.2 Transfer Heat Efficiently

After power is consumed and heat is generated, the challenge then becomes how that thermal energy is transferred to ambient. Methods and apparatuses for heat transfer are being created in abundance. The field of heat transfer and the techniques used to solve heat transfer problems can be categorized into three pieces:

1. *Conduction*. The transfer of energy between objects that are in physical contact (heat sinks, potting, thermal tape).
2. *Convection*. The transfer of energy between an object and its environment, due to fluid motion (air, fluid).
3. *Radiation*. The transfer of energy to or from a body by means of the emission or absorption of electromagnetic radiation.

Conduction is the transfer of heat between substances that are in direct contact with each other. The better the conductor, the better heat will transfer. Examples of good conductors are aluminum, copper, silver, iron, and steel. Examples of bad conductors (insulators) are plastic, wood, paper, and air. In embedded systems, it's important for materials to be in contact with each other in order for heat to transfer efficiently. For processors, this can be underneath the part where it makes contact with the PCB using a thermal pad, or on top of the part if it can be connected with a heat sink, a thermal pad, or a connection with the mechanical case for the device. Each of these methods will allow heat to conduct through the materials and outside to ambient as long as the thermal resistance of the material is less than air (which is easy to do).

Convection is the up-and-down movement of gases and liquids caused by heat transfer. As a gas or a liquid is heated, it becomes less dense and rises. When a gas or a liquid cools, it becomes more dense and falls. These movements constitute a convection current by which heat can move. In embedded systems, convection can be facilitated through the use of air movement, or liquid movement. Since liquids are better at transferring heat than air, a liquid-cooled thermal system is better than an air-based system. However, this is usually not an option for embedded systems due to the cost and space requirements.

Thermal radiation is the process by which electromagnetic waves travel through space. When the waves come into contact with an object, the waves transfer heat to that object.

When an electronic system heats up, there is a certain rate at which it heats, given by factors such as thermal mass, power run through the system, efficiency of the system, etc. After the system heats up, there is then a cool-down curve that is affected by things such as thermal mass, surface area, rate of convection, ambient temperature, rate of conduction, and others.

How can software help here? A few ways:

- Programmatically control heat spreading equipment such as fans, or blades, or motors controlling the rate of flow of a liquid cooling system.
- Aggressively reduce power after peak power points. As the system is cooling down, de-prioritize follow-on processing requests to give the system time to cool down. This approach is also sometimes called time-coalescing.

Summary

- There are three ways to transfer heat: conduction, convection, and radiation.
- Software can assist with heat transfer by controlling fans, blades, or motors that drive an electrical-mechanical thermal management system if it exists.
- Software can also help with heat transfer by increasing the duration of time between peak power consumption and heat production events. This is sometimes referred to as time-coalescing.

2.3.3 Define The Environment

The third approach to managing heat in embedded systems is to define the environment. If it can be guaranteed that the device will only operate within certain constraints (inside a cooler, in Canada, not in an oven, etc.), the job of transferring heat can be made easier. Note that if the ambient temperature in the environment is high, the difference between the junction temperature in the integrated circuit under load will be small, and hence the rate of heat transfer will be small. Conversely, if the ambient temperature is low, the difference between the junction temperature and the environment will be large, and thus the rate of heat transfer will be high.

For software, this approach is the one that is least controllable unless the software is in control of thermal systems within the environment. For example:

- As power consumption increases, control external cooling elements so the environmental temperature decreasing, encouraging heat transfer to happen more quickly.

- Tell the user that danger of overheating is present and ask the user to assist in controlling the environment, or cease using the equipment for a period of time.

The landscape of SOFTWARE THERMAL MANAGEMENT is rich and growing. As our computing requirements increase, so will our need to manage thermal performance with software, especially for applications that require high computation in short periods of time such as with multimedia embedded systems.

Summary

- A third way to manage the thermal performance of an embedded system is to define the environment.
- The greater the difference between the embedded system and ambient temperature, the faster heat will transfer.
- Software can affect the environmental thermal impact to an embedded system if the software can control the environmental temperature or instruct a user to perform a task that increases the rate of heat transfer (turn on a fan, reduce ambient temperature, or cease to use a device for a period of time).

2.4 Crossroads

The field of STM is really an intersection of three fields: Thermodynamics, Electrical Engineering, and Software Engineering, as shown in Fig. 2.16.[3]

Software engineers play a major part in reducing power consumption for embedded systems, but are not required by universities to take courses on thermodynamics, or complex power-management circuit design. The aim of the field of SOFTWARE THERMAL MANAGEMENT is to encourage software engineers to play a more active and central role in the task of managing the thermal performance of embedded systems.

Summary

- The field of STM sits at the intersection of Thermodynamics, Electrical Engineering, and Software Engineering.

[3] The field of SOFTWARE THERMAL MANAGEMENT is similar to the concept of software power management, except that it is narrower in scope, and focuses on thermal performance of the system instead of broadly looking at system-wide power draw.

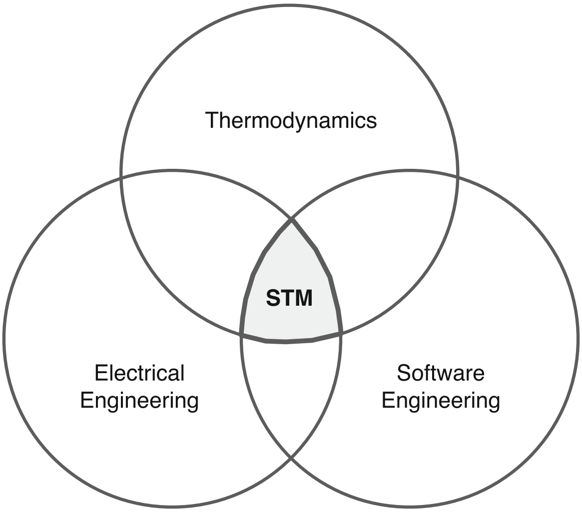

Fig. 2.16 SOFTWARE THERMAL MANAGEMENT sits at the intersection of Thermodynamics, Electrical Engineering, and Software Engineering. Software engineers play a major part in reducing power consumption for embedded systems, but are not required by universities to take courses on thermodynamics, or complex power-management circuit design. The aim of the field of SOFTWARE THERMAL MANAGEMENT is to encourage software engineers to play a more active and central role in the task of managing the thermal performance of embedded systems

- Software engineers play a central role in the amount of power consumed by an embedded system, and therefore should be encouraged to actively develop frameworks and techniques to limit peak thermal events, and lower the overall power consumption of the system without sacrificing key user scenarios.

2.4.1 Thermodynamics

Thermodynamics is a sub-field of physics that deals with the relationship between heat and other properties of substances such as pressure and temperature.

In particular, thermodynamics focuses on how heat is transferred and is related to energy transformation within a physical system undergoing a thermodynamic process. These types of processes result in work being done by the system, and are guided by the laws of thermodynamics.

In the field of thermodynamics, and particularly with the study of heat transfer, there is much material, academic papers, conferences, journals, research departments (Villanova Thermal Management Laboratory, Stanford NanoHeat Laboratory), and companies that have their foundations in physics and mechanical engineering and are focused intently on how to transfer heat effectively and efficiently.

For example, The Journal of Thermal Science from Springer Verlag, edited by Shen Yu, offers the following mission: “Journal of Thermal Science publishes high-quality articles on experimental, numerical and theoretical investigations which give insight into the major areas of thermal and fluid sciences. It publishes contributions in the fields of fluid mechanics, aerothermodynamics [...], heat and mass transfer, multiphase flow, turbulence modeling, combustion, engineering thermodynamics, thermophysical properties of matter, measurement and visualization techniques”.

As the needs for heat transfer solutions have grown, there has also emerged follow-on industries that have developed in response. For instance, with convection, moving air is solved typically by the addition of fans, or with sophisticated fluid cooling systems. As convection and convection/conduction systems such as these operate, the fans or motors that control the thermal management system operate, produce noise. The resulting noise problem is then solved by techniques and methods for making fans that operate smoother and more efficiently, or even by adding acoustic dampening material, or active-noise cancellation techniques.

For military applications, if we look at the list of supported projects through the DARPA Thermal Management Technologies (TMT) program, we find similar themes. The overarching goal of the TMT program is to “explore and optimize new nanostructured materials and other recent advances for use in thermal management systems” and presently includes the following areas of focus:

- **Thermal Ground Planes (TGP).** The TGP effort is focused on high-performance heat spreaders which use two-phase cooling to replace the copper alloy spreaders in conventional systems.
- **Microtechnologies for Air-Cooled Exchanges (MACE).** The goal of the MACE effort is to enhance air-cooled exchangers by reducing the thermal resistance through the heat sink to the ambient, increasing convection through the system, improving heat sink fin thermal conductivity, optimizing and/or redesigning the complimentary heat sink blower, and increasing the overall system (heat sink and blower) coefficient of performance.
- **Nanothermal Interfaces (NTI).** The NTI effort is focused on novel materials and structures that can provide significant reductions in the thermal resistance of the thermal interface layer between the backside of an electronic device and the next layer of the package, which might be a spreader or a heatsink. ACM will investigate active cooling of electronic devices using techniques such as thermoelectric coolers, sterling engines, etc.
- **Near Junction Thermal Transport (NJTT).** The goal of the Near Junction Thermal Transport (NJTT) effort of the TMT program is to achieve a 3x or greater improvement in power handling from GaN power amplifiers through improved thermal management of the near junction region.

These technology focuses are all related to the process of heat transfer, as opposed to reducing the sources of heat.

There are also industry awards given out for contributions to the field of thermal management. Each year, at SEMI-THERM (a conference on "thermal innovations that make the world's technology cool"), gives a THERMI Award to one person who has made significant contributions to the field of thermal science. This conference, and the work that is highlighted and rewarded in this conference and others like it are related to heat transfer (e.g. microfluidic cooling systems for high performance PCs).

For embedded systems, transferring heat is important, and the work that is being done by physicists, materials scientists, and chemists is critical. Electrical engineers are involved in the study of heat transfer to the degree that they are involved in how integrated circuits are designed and operate. However, software engineers typically are not part of the discussion for heat transfer or thermodynamics since software engineers do not typically deal with physical materials, thermal resistances, or the design of electronic components.

In order for software engineers to participate in thermal management of an embedded system, it's important that software engineers understand the basics of thermodynamics. An overview of thermodynamics for software engineers is given in Sect. 3.2.

Summary

- Thermodynamics guides the fundamental behavior of the universe, including the process of power dissipation in embedded systems.
- The field of thermodynamics focuses on the study of energy transfer, and software engineers are not usually required to take courses on this topic.
- Software engineers play a critical role in thermal management, but are often not part of the discussion. In reality, software engineers play a critical role in the thermal performance of embedded systems since software controls the amount of computation is used, power is consumed, and heat is produced.

2.4.2 Electrical Engineering

Electrical engineering is a field dedicated to the design, development, building and testing of electronic devices such as TVs, embedded computer systems, generators, microprocessors, and amplifiers. Electrical engineering studies in universities focus on the fundamentals of electricity and magnetism, circuits, printed circuit board layout, and manufacturing processes for electronic devices.

Power efficiency and power management is an active area of research, and has been for some time. Numerous sources exist on this important topic. [16–27, 31, 32] In embedded systems there is a science to choosing the right processor for a

Table 2.2 Common microcontroller selection parameters are given here. Picking a microcontroller can be difficult

Parameter	Description
Architecture	ARM, Atmel AVR, Microchip PIC, TI MSP430, etc.
Price	Inversely proportional to unit volume commitments
Package type	Defines how the packaged is mounted to the printed circuit board
CPU speed	Quadratic relationship to power consumption
RAM	Volatile storage capacity
Flash	Non-volatile read/write storage capacity
EEPROM	Another type of erasable non-volatile storage
Operating voltage range	The minimum and maximum voltage this processor can handle
Temperature range	The temperature range, outside of which causes undefined behavior
I/O pins	Input and output pins that are available
Timers	For setting and measuring time-based events
ADC channels	Converts analog voltage levels to digital values
CODECs	Algorithms for converting or filtering data streams in hardware
DAC modules	For converting digital values to analog voltages
DMA channels	Allows independent memory access without interrupting the CPU
RTC	Real time clock
I2C	Multimaster serial bus for interfacing with low speed peripherals
I2S	Serial bus interface for connecting digital audio devices together
IrDA	Protocol support for infrared communications
PWM outputs	Modulates pulse width to encode data or fine-tune delivery of power
SPI	Serial interface for bi-directional communication with peripherals
UART	Serial port, one or many
USB	Universal serial bus for device and/or host modes

However, with the help of parametric searches, the process can be made more easily. Even with parametric searches, however, the art of choosing a processor that will meet thermal performance requirements in a system is rather subjective, and hard to quantify at the beginning of a program, when electrical components are being selected

given design. Usually processors are chosen based on objective factors such as price, peripheral support, physical packaging, memory, architecture family, and software tools available.

With microcontrollers, it's common for a vendor to offer a parametric search that allows product designers to narrow the wide array of options to choose a processor that fits their needs for their specific product design. Some of the more common parameters for choosing a microcontroller are listed in Table 2.2.

Picking a processor is not always an easy task, but with the help of vendor-specific parametric searches, and application notes as a guide, it can be done in a deductive and deterministic way.

Picking a part that will meet you thermal and power goals, on the other hand, can be tricky. Power consumption and thermal behavior can be relatively simple with small 4- and 8-bit microcontrollers, but for advanced 32-bit microcontrollers that also process video and audio on demand, they can be extremely powerful, but also difficult to understand and model.

The data sheets that accompany such a processor can be hundreds of pages to many thousands of pages of information on how to power, configure, and manage that processor in a given design. The technical reference manual for the TI OMAP4470, for instance, is 5,000+ pages. With this much information and levels of configuration within the processor, it's nearly impossible to make a quantitative determination with high certainty that a given processor will meet the thermal requirements of the system.

If thermal performance will be an important issue for a given design, here are some key questions to ask when selecting a processor:

1. How low can the processor go (power draw) while in an idle state? How much work is required to get there?
2. How does this change when the processor is under high load such as when it may be decoding video streams?
3. How much heat will the processor create during a specific use cycle of off, idle, running, encoding/decoding audio/video?
4. If the processor gets too hot, can it be turned off or scaled back? Will it turn itself off if it gets too hot? Can I run in a lower mode for a period of time until it cools off?

These questions are difficult to answer at the outset of a project and is why SOFTWARE THERMAL MANAGEMENT should be considered both a science and an art.

Silicon vendors such as Texas Instruments, Freescale, Intel, AMD, Nvidia, and Qualcomm provide a wealth of thought-leadership on the topic of software power management, power scaling, and also thermal design considerations. This is due to the fact that these silicon vendors manufacture processors that are well-suited for high-performance embedded systems which need to be fully operational when the users are viewing video, streaming media, listening to audio, or processing real-time data streams, but go into very low power modes when idle to conserve precious battery capacity (and thermal output).

Summary

- Electrical engineers play a major role in designing electronic systems to meet thermal performance requirements.
- Picking a processor is a difficult task when thermal performance is important. It's difficult at the early stages of a program to quantitatively model and predict thermal performance such that one can know with high certainty that a given processor is *the right one*.
- Circuit design surrounding the processor selection also has a major impact on thermal performance. By designing the system to have many options for gating power to key circuits in the design, and by laying out the circuit board (and accompanying thermal pads, vias, and board layers) to transfer heat efficiently, they have a major role to play in enabling software engineers to control the overall thermal performance of the end system.

2.4.3 Software Engineering

The term *software engineering* has only been around since the late 1950s. It's a relatively new profession that is concerned with organizing and facilitating computation into meaningful work and user interactions.

Compared to physics, the field of software engineering is infantile, and the processes and techniques used to create complex software systems are varied and colloquial.

Software engineers develop software to solve problems, and hopefully do so in a way that is reliable, efficient, maintainable, safe, secure, usable, fast, and satisfy all the requirements that are defined for the system.

Often, there are trade-offs to make, such as whether power or performance is more important. In embedded systems, software engineers have additional responsibility to make sure that their code is compact and fits within the available code storage, and is power-efficient since the microprocessors that are used are not nearly as powerful or capable as their desktop counterparts, and often include batteries for power supplies that must be used sparingly.

In the field of SOFTWARE THERMAL MANAGEMENT, hardware features are critical to the ability to manage power and heat in a system. However, the entire system will not work, nor will it meet its thermal and power consumption goals without the aid and control of software algorithms.

The role of software software is to choreograph the hardware at the right time and in the right way based on the needs of the system, the needs of the user using the system, and to protect against safety or regulatory risks. See Fig. 2.17.

In the world of software engineering, power management is an important and respected topic of concern. The concern, however, is usually with regard to the broader topic of system-level power management to conserve battery power, save on energy costs, or to reduce environmental impact.

However, the field of SOFTWARE THERMAL MANAGEMENT is to focus in on the narrower topic of thermal performance, and intends to reduce moments of peak thermal impact so as not to incur unnecessary thermal fatigue, and to reduce the risk of failure. This focus on thermal performance is a subset of power management. In order to do it well, software engineers must have some knowledge of hardware design and thermodynamics. These topics are discussed in the coming chapters.

Summary

- Software engineering is a new field when compared to Classical Thermodynamics. The number and variation of types of software development processes, patterns, refactoring techniques, and architectural styles are high.
- The field of SOFTWARE THERMAL MANAGEMENT sits at the intersection of Thermodynamics, Electrical Engineering, and Software Engineering.

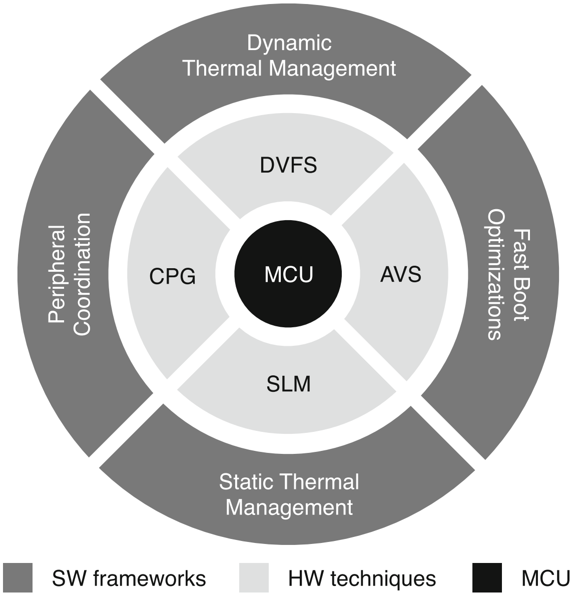

Fig. 2.17 Software plays a special role in thermal management architectures. Starting with the processor, and the features supported by the hardware, it's up to the software to determine how and when to switch to and from low-power modes. In the center of the diagram is the Microcontroller Unit (MCU). Categories of power and thermal features supported by the hardware include Dynamic Voltage and Frequency Scaling (DVFS), Adaptive Voltage Scaling (AVS), Static Leakage Management (SLM), and Clock and Power Gating (CPG) features

- As processors and other hardware is becoming more sophisticated, the potential for high computation in embedded systems is also increasing. It's up to software engineers to coordinate hardware resources such that the use cases of the system are satisfied, and to do so in a way that uses the minimum amount of power as possible.
- Software engineers, in order to be experts in the field of SOFTWARE THERMAL MANAGEMENT, must have a base working knowledge of hardware design and thermodynamics.

References

1. Allen, B.: Information Tasks: Toward a User-Centered Approach to Information Systems. Academic Press Inc, Orlando (1996)
2. Kuniavsky, M.: Observing the User Experience: A Practitioners Guide to User Research. Morgan Kaufmann, Burlington (2003)

3. Blom, J., Chipchase, J., Lehikoinen, J.: Contextual and cultural challenges for user mobility research. Commun. ACM. **48**, 37–41 (2005)
4. Kumar, V., Whitney, P.: Faster, cheaper, deeper user research. Des. Manag. J. (Former Series). **14**, 50–57 (2003)
5. Brittain, J.M.: Pitfalls of user research, and some neglected areas. Soc. Sci. Info. Stud. **2**, 139–148 (1982)
6. Cooper, A., Reimann, R., Cronin, D., Cooper, A.: About Face 3: The Essentials of Interaction Design. Wiley Pub, Indianapolis (2007)
7. Norman, D.A.: The Design of Everyday Things. Basic Books, New York (2002)
8. Krug, S.: Dont Make me Think!: A Common Sense Approach to Web Usability. New Riders Pub, Berkeley (2006)
9. Goodwin, K.: Designing for the Digital Age: How to Create Human-Centered Products and Services. Wiley Pub, Indianapolis (2009)
10. Cooper, A.: The Inmates are Running the Asylum. Sams, Indianapolis (2004)
11. Saffer, D.: Designing for Interaction: Creating Innovative Applications and Devices. New Riders; Pearson Education [distributor], Berkeley (2010)
12. Bowles, J.B.: A survey of reliability-prediction procedures for microelectronic devices. IEEE Trans. Reliab. **41**, 212 (1992)
13. Marchionini, G.: Information Seeking in Electronic Environments. Cambridge University Press, Cambridge (1997)
14. Ohring, M.: Reliability and Failure of Electronic Materials and Devices. Academic Press, Boston (1998)
15. Grimm, R., Anderson, T., Bershad, B., Wetherall, D.: A system architecture for pervasive computing. Proceedings of the 9th Workshop on ACM SIGOPS European Workshop: Beyond the PC: New Challenges for the Operating System, pp. 177–182. ACM, New York (2000)
16. Klauk, H., Zschieschang, U., Pflaum, J., Halik, M.: Ultralow-power organic complementary circuits. Nature **445**, 745–748 (2007)
17. Von Kaenel, V.R., Pardoen, M.D., Dijkstra, E., Vittoz, E.A.: Automatic adjustment of threshold and supply voltages for minimum power consumption in CMOS digital circuits. IEEE Symposium on Low Power Electronics 1994. Digest of Technical Papers, pp. 7879 (1994)
18. Soeleman, H., Roy, K.: Ultra-low power digital subthreshold logic circuits. Proceedings of the 1999 International Symposium on Low Power Electronics and Design, pp. 94–96. ACM, New York (1999)
19. Hemani, A., Meincke, T., Kumar, S., Postula, A., Olsson, T., Nilsson, P., Oberg, J., Ellervee, P., Lundqvist, D.: Lowering power consumption in clock by using globally asynchronous locally synchronous design style. Proceedings of the 36th ACM/IEEE Conference on Design automation 1999, pp. 873–878 (1999)
20. Kim, N.S., Austin, T., Baauw, D., Mudge, T., Flautner, K., Hu, J.S., Irwin, M.J., Kandemir, M., Narayanan, V.: Leakage current: Moores law meets static power. Computer **36**, 68–75 (2003)
21. Erickson, R.W., Maksimovic, D.: Fundamentals of Power Electronics. Springer, Netherlands (2001)
22. Girard, P., Landrault, C., Pravossoudovitch, S., Severac, D.: Reduction of power consumption during test application by test vector ordering [VLSI circuits]. Electron. Lett. **33**, 1752–1754 (1997)
23. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) Advances in Cryptology CRYPTO 99. pp. 388–397. Springer, Berlin (1999)
24. Maksimovic, D., Zane, R., Erickson, R.: Impact of digital control in power electronics. Proceedings of The 16th International Symposium on Power Semiconductor Devices and ICs, 2004 (ISPSD 04), pp. 13–22 (2004)
25. Ye, T.T., Benini, L., De Micheli, G.: Analysis of power consumption on switch fabrics in network routers. Proceedings of the 39th Design Automation Conference 2002, pp. 524–529 (2002)
26. Hicks, P., Walnock, M., Owens, R.M.: Analysis of power consumption in memory hierarchies. Proceedings of the 1997 International Symposium on Low Power Electronics and Design, pp. 239–242. ACM, New York (1997)

27. Piguet, C.: Low-Power Electronics Design. CRC Press, Boca Raton (2004)
28. Moore, G. E.: Cramming more components onto integrated circuits. Electronics **38** (8) (1965)
29. Sutter, H.: The free lunch is over: a fundamental turn towards concurrency in software. Dr. Dobb's J. **30**(3) (2005)
30. I.T.: IEC 60134 Ed. 1.0 b:1961, Rating systems for electronic tubes and valves and analogous semiconductor devices. Multiple. Distributed through American National Standards Institute (2007)
31. Kaxiras, S., Martonosi, M.: Computer Architecture Techniques for Power-efficiency. Morgan & Claypool Publishers, Seattle (2008)
32. Rabaey, J.M.: Low Power Design Essentials. Springer, New York (2009)

http://www.springer.com/978-1-4939-0297-2

The Art of Software Thermal Management for
Embedded Systems
Benson, M.
2014, XVI, 124 p. 51 illus., 2 illus. in color., Hardcover
ISBN: 978-1-4939-0297-2