

Increasing the Sensitivity of On-Chip Digital Thermal Sensors with Pre-Filtering

Zhimin Chen, Raghunandan Nagesh, Anand Reddy and Patrick Schaumont
ECE Department, Virginia Tech
{chenzm, magesh, anandr, schaum}@vt.edu

Abstract

Thermal monitoring has been broadly used to protect high-end integrated circuits from over-heating and to identify hot-spots in complex circuits. In this paper, we present a method to increase the sensitivity of an on-chip digital thermal sensor. In contrast to the existing mechanisms that characterize the overall temperature profile on a die, our solution is able to detect the submerged thermal variation caused by specific predefined events (SPE), under the precondition that the SPE's major frequency component does not overlap with those of other thermal events. This is made possible by pre-filtering of the temperature value. A demonstrator is implemented in an ordinary FPGA, in which the SPE is a person's finger touching on the FPGA package. We successfully show that our design can do a correct and reliable detection of the finger touching event while ignoring other larger variations caused by other reasons. Because the finger touching event has no other special characteristics except for its unique frequency, we conclude that our solution is also applicable to other SPEs, especially low-frequency ones. In general, our method is sensitive, reliable and also flexible.

1. Introduction

Thermal sensors have been broadly used to monitor high-end chips because the high circuit density results in an increased power dissipation density [1] [2]. This makes the die temperature to reach a level where it could damage the chip. Another application of thermal sensors is to detect the hot-spots of a circuit [3]. This application integrates on-chip thermal sensors (usually a sensor grid) into a circuit under test, to detect, and to compare local temperature. In addition, recently, another interesting application for thermal detection has been presented, called 'passive thermal tag' [4]. The designers show that passive thermal tag can be

used to build up a covert thermal communication channel, which is useful to support intellectual-property protection protocols.

Thermal monitoring thus has a large range of useful applications, and thermal sensors are essential to collect temperature data on a design. In this paper, we investigate the problem of increasing the sensitivity of a temperature sensor. The sensitivity here is defined as the capability to sense temperature variation caused by *specific pre-defined events* (SPE). The SPE can be a specific heat source or even the specific activity of a heat source. In this paper, *what differentiates one SPE from other thermal events is its major frequency component.*

A more sensitive thermal sensor could be used to monitor the activity of a specific heat source and therefore distinguish it from others. It can also be used for on-line thermal testing. All of these have not been achieved with the existing thermal sensors.

It is not easy to solve the above problem. One of the main reasons is that the existing thermal sensors only monitor the overall temperature variation profile. If we need to observe the temperature variation caused by specific events, its amplitude should be comparatively large, so that it does not merge into the overall temperature change. However, a single thermal event in a large system only contributes a small part to the overall temperature. Its influence is usually compensated or surmounted by other thermal events. Accordingly, simply observing the amplitude of the overall thermal variation profile, which is done by the existing thermal sensors, cannot fulfill our purpose.

This paper presents our first step to solve this problem. The main idea is to focus on the SPE with a band-pass frequency filter. Our contribution contains a detailed analysis of a ring-oscillator based thermal sensor in the frequency domain. We show that a counter inside the thermal sensor acts as a low-pass filter. However, because the counter is periodically sampled, it also introduces a negative effect called aliasing. Based on the analysis, we propose a solution

in which the band-pass filter is built up with 3 parts: the counter based low-pass filter, the low-pass post-processing, and the high-pass post-processing. This solution not only makes use of the low-pass feature of the counter, but also reduces its negative effect. Finally, the solution is simple and flexible. As far as we know, there has been no similar discussion before. In addition, we designed a demonstrator in an ordinary FPGA. The design can monitor a low frequency SPE – a finger touching on the FPGA package – correctly and reliably.

The rest part of this paper is organized as follows. Section 2 goes through some preliminaries on thermal detection. In Section 3, we present our solution for sensitive and stable thermal detection. A prototype design is described in Section 4. Section 5 shows the experimental results and finally Section 6 concludes this paper.

2. Preliminaries

2.1. Ring-oscillator based thermal sensor

An ordinary on-chip digital thermal sensor consists of a ring oscillator, a counter clocked by the ring oscillator and a sampling register.

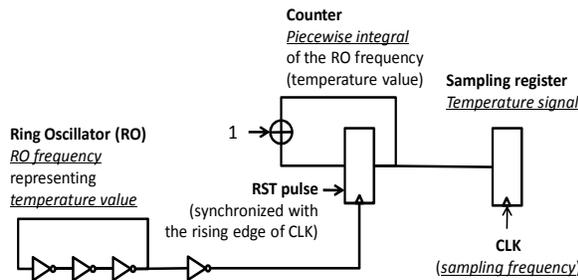


Figure 1: Ring-oscillator based thermal sensor structure

As we can see from Fig. 1, the ring oscillator is a series of odd number of invertors connected in a ring format. The *ring oscillator frequency* represents the *temperature value*. To collect the information of the temperature value, a counter is used to count the 0-to-1 transitions at the ring oscillator output [5]. Periodically, we sample and reset the counter. Those sampled counter values are the *temperature samples*, and the counter sampling rate is defined as *sampling frequency*. The count value is an integral of the ring oscillator frequency. For each sampling period, the counter starts from 0 because of the reset. The counting operation thus is a *piecewise integral*. The main advantage of the ring oscillator based thermal sensor is its easiness for integration. Because of this reason, the majority thermal sensors in FPGA fall into this type [6]

[7]. This is also the reason why we choose ring oscillator based thermal sensors in our design.

2.2. Frequency of the temperature value

In this section, we try to analyze the frequency spectrum of the temperature value in the frequency domain.

There are many possible thermal events that may have influence on the temperature value, such as environmental temperature variation, activity of other circuit modules on the same chip, and noise. *Different thermal events usually have different major frequencies*. For example, the variation of the environmental temperature normally takes hours and thus has a low-frequency component; the UART circuit module operating at the baud rate of 9600 has its major frequency component between 500Hz to 1KHz. The individual thermal events can be approximately added up to form the overall influence, and the resulting temperature value includes similar frequency components as the individual events.

Under the *precondition* that the SPE's frequency is different from other events' frequencies, a reasonable hypothesis is that, in the spectrum of the temperature values, the frequency component from SPE is usually higher than the one from environmental air temperature variation and also different from those from other circuit modules. Moreover, the energy of noise within a small frequency range is small, so it has a small influence on the band-limited SPE-caused frequency component.

The above analysis tells us that it is possible to observe the activity of the SPE by focusing on its major frequency component while filtering out others. As a result, *a band-pass filtering on the temperature value can satisfy the requirements*, which will be detailed in the next section.

3. Our solution

In this section, we present our solution to enable a thermal sensor to monitor the activity of the SPE based on the band-pass filtering on the temperature value. We choose a normal ring oscillator based thermal sensor as the starting point. Based on that, we present our effort to make the solution correct and flexible.

In Section 2.1, we showed that a counter clocked by the ring oscillator performs a piecewise integral of the ring oscillator frequency. Since the integral already forms an existing low-pass filter, we decide to divide the band-pass filter into a low-pass filter and a high-pass filter in order to make use of the existing resources.

3.1. Low-pass filter

Because the temperature value is represented by the ring oscillator frequency, they have similar frequency spectrums. The following analysis is made on the ring oscillator frequency.

Let's first look at the counter based low-pass filter. Suppose the sampling period is T_s , the frequency of the ring oscillator is $freq(t)$, then the sampled counter value $C(n)$ (also called temperature samples) can be represented in Eq. 1.

$$C(n) = \int_{(n-1)T_s}^{nT_s} freq(t) dt \quad (1)$$

The operation in Eq. 1 includes two steps: 1) a continuous integral; 2) converting the continuous-time result of step 1 to a digital signal by sampling. *The sampling period in step 2 always equal to the integral length in step 1.* We represent these two steps as follows in Eq. 2 and Eq. 3.

$$I(t) = \int_{t-T_s}^t freq(t') dt' \quad (2)$$

$$C(n) = I(nT_s) \quad (3)$$

Eq. 2 is an integral function in the time domain. Its transfer function $H(f)$ in the frequency domain is $T_s * Sinc(T_s * f)$ [7], which is shown in Fig. 2. Then the amplitude of $I(t)$'s Fourier transform $I(f)$ is

$$\begin{aligned} |I(f)| &= |Freq(f)H(f)| \\ &= |Freq(f) * T_s * Sinc(T_s * f)| \end{aligned} \quad (4)$$

$Freq(f)$ is the Fourier transform of $freq(t)$. In Fig. 2, we can see a large main lobe from $-f_s$ to f_s as well as a lot of smaller side lobes. Most of the frequency components of $Freq(f)$ in the main lobe will be preserved. All the others are depressed to a large extent. Therefore, the integral operation acts like a low-pass filter. For analysis convenience, we approximate the highest frequency that can pass this low-pass filter to be f_s .

The following sampling operation brings us some problems. We can see that the output of Eq. 2 is sampled in Eq. 3 with f_s as the sampling frequency. But according to the Sampling Theorem [9] [10], the sampling frequency should be at least $2*f_s$. Otherwise, we cannot recover the sampled signal because of aliasing. In detail, in the frequency domain, sampling duplicates the $I(f)$ and shifts the copies by $n*f_s$ ($n = \dots -1, 0, 1, \dots$), and finally adds them up. The spectrum of $C(n)$ is presented in Eq. 5.

From Eq. 5, we can see how *aliasing* occurs. Suppose the major frequency of the SPE is f_m , after sampling, it will overlap with another frequency component previously located at $f_i = f_m - f_s$. As an

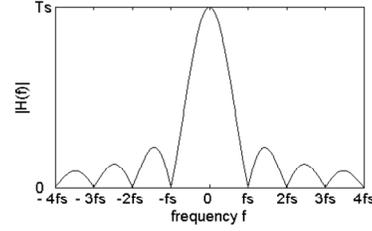


Figure 2: Amplitude of the integral transfer function

$$C(f) = \sum_{n=-\infty}^{\infty} Freq(f + n * f_s) Sinc(T_s(f + n * f_s)) \quad (5)$$

example, we assume the temperature value has a uniform frequency distribution ($Freq(f) = 1$). Then after the integral operation, $I(f) = H(f)$. The sampling effect is shown in Fig. 3(a). Before sampling, the frequency component at f_i is represented by the dotted arrow. After the sampling, one of its copies is shifted to the position of f_m . We can see that the most serious aliasing exists when the neighboring main lobes overlap with each other by a frequency range of f_s . Here, we neglect the influence of side-lobes.

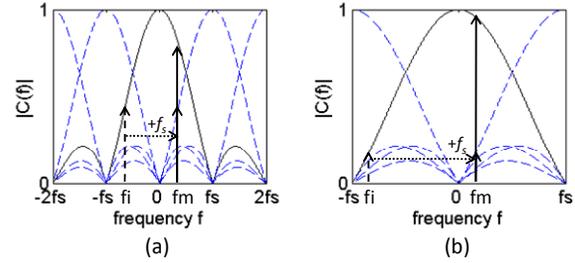


Figure 3: (a) Aliasing occurs after sampling; (b) Influence on f_m of aliasing decreases as f_s increases.

An important point is that the sampling period always equals to the integral range (an intrinsic characteristic of the counter). This means that the aliasing is *unavoidable!* To reduce the influence of f_i , our solution is to increase f_s , which is shown in Fig. 3(b), where the sampling frequency is two times the one in Fig. 3(a). When f_s increases, $|f_i|$ increases accordingly. The higher the $|f_i|$ is, the smaller $Sa(f_i)$ becomes. That is why we see a smaller intensity at frequency f_i in Fig. 3(b). Meanwhile, f_m remains the same, and $Sa(f_m)$ becomes larger as f_s increases. The ratio of $Sa(f_i)$ over $Sa(f_m)$ becomes smaller, so the aliasing influence is reduced.

Choosing f_s is a tradeoff. A higher f_s leads to less error caused by aliasing. However, it also causes higher cut-off frequency of the counter based low-pass filter. More noise is kept in the filter's output. In addition, higher f_s also requires a higher cost for the following post-processing.

To improve the effect of low-pass filtering, we introduce a post-process based on the above temperature samples. This process can be as complicated as a low-pass filter or as simple as another integral. In our experiments, we use the latter. The process can be represented as follows where *sample* is the temperature sample and *sum* is the process result.

$$width = f_s / (2 * f_m);$$

$$sum[i] = sample[i - width + 1] + \dots + sample[i];$$

This is similar to the above piecewise integral based on the counter. The difference is that the integral here is performed in a moving window way not piece by piece. Hence, the sampling frequency does not change according to the integral range.

With this moving window integral, the temperature samples go through a low-pass filter with a cut-off frequency close to $2 * f_m$. Considering the similarity of integral operations, we do not cover the details.

Based on the above analysis, we can conclude Section 3.1 as follows. The counter based low-pass filter guarantees to filter out most of the frequency components of the temperature value that is higher than f_s . Increasing f_s also guarantees that the aliasing phenomenon has a small influence on the frequency component f_m from SPE. The post-processing filters out most of the unwanted frequency components between $2 * f_m$ and f_s . By combining both of them together, we have reached the point that most of the frequency components that are higher than $2 * f_m$ have been blocked. The low-pass filtering's purpose has been fulfilled.

3.2. High-pass filter

Unlike the low-pass filter, which has the counter as an existing low-pass resource, there is no existing resource for the high-pass filter. The high-pass filter has to be implemented as a post-process. To make our solution as simple as possible, we use a simple differential operation to implement high-pass filter as follows.

$$width = f_s / (2 * f_m);$$

$$diff_1st[i] = sum[i - width + 1] - sum[i];$$

$$diff_2nd[i] = diff_1st[i - width + 1] - diff_1st[i];$$

sum is the result from the low-pass post-process and *diff_2nd* is the output of high-pass post-process. The transfer function of this high-pass filter is shown in Eq. 6.

$$H(f) = 1 - 2 * e^{-j2\pi f * width} + e^{-j2\pi f * 2 * width} \quad (6)$$

After the low-pass filtering, the frequency components that are above $2 * f_m$ are mostly filtered out,

so we only look at the frequency range from 0 to $2 * f_m$ here.

As we can see from Fig. 4, the high-pass filter's transfer function $|H(f)|$ starts from 0 and increases to around 4 as the frequency increases from 0 to $2 * f_m$. Obviously, higher frequency components in this range can pass the filter more easily. We find this filter is sufficient for our experiment.

According to the above discussion, we have already built up a low-pass filter and a high-pass filter. The low-pass filter filters out most of the frequency components that are higher than $2 * f_m$. The high-pass filter blocks most of the frequency components that are lower than f_m . In combination, these two filters form a band-pass filter. This helps us to focus on SPE's frequency component f_m and finally, to enable the thermal sensor to monitor the activity of the SPE. (If there is an unwanted frequency component that is close to f_m , the post-processing should be more complicated. We do not discuss this in this paper.)

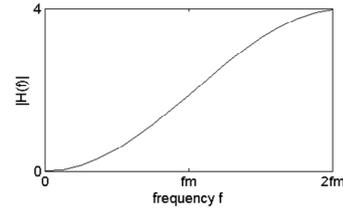


Figure 4: Amplitude of H(f)

4. The design

In this section, we present a design in an FPGA according to the solution in Section 3. The purpose of the design is to detect a low-frequency SPE: a person's finger touching on the FPGA package. The temperature difference between the finger and the FPGA package will lead to a heat transmission, and hence, influence the internal thermal sensor. Based on the experimental results, we find it usually takes around 20 seconds for the FPGA to return to the original status after a finger touching. Therefore, the main frequency (f_m) of the finger touching signal is around 0.025Hz. The structure of the design is depicted in Fig. 5.

Inside the thermal sensor, the sampling frequency (f_s) on the counter is set to 2.98Hz (nearly 120 times of the f_m). The post-processing is done within PicoBlaze, an 8-bit micro-controller, which runs at 50MHz.

Besides that, for the experimental purpose, PicoBlaze also transfers intermediate and final results to the PC through the RS232 cable. The post-processing program for PicoBlaze is presented as follows in Fig. 6.

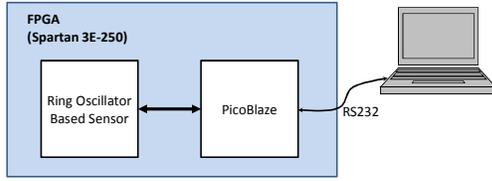


Figure 5: Structure of the design

```

initialize sa_p, su_p, d1_p and i to 0;
set width to 60;
while(new temperature sample is available)
    new_sum=sum[su_p]-sample[sa_p]+new_sample;
    sample[sa_p]=new_sample;
    sa_p=(sa_p+1)%width;
    new_diff_1st=new_sum-sum[su_p];
    sum[su_p]=new_sum;
    su_p=(su_p+1)%width;
    diff_2nd=new_diff_1st-diff_1st[d1_p];
    diff_1st[d1_p]=new_diff_1st;
    d1_p=(d1_p+1)%width;
    i=i+1;
    if(i>3*width)
        send diff_2nd to PC;

```

Figure 6: Program in PicoBlaze

5. Experimental results

The experimental setup consists of a personal computer (PC) and a Digilent Basys Board [11] with a Xilinx Spartan 3E-250 FPGA. The computer receives data from FPGA through an RS232 cable. Inside the FPGA, the design presented in Section 4 is included. Besides that, another design running at 50MHz is also implemented. 92% of the total slices are occupied.

The experiments were done in an ordinary indoor environment where the surface temperature of the FPGA package is within 20 – 30 °C , and the temperature of a finger is around 37°C. During the experiments, we press our thumb on the FPGA for 2-3 seconds. Results show that every finger touching can be detected by the FPGA.

In the rest of this section, we present the results of one of the experiments. In this experiment, the sampling frequency f_s is 2.98Hz. The major frequency of the finger touching signal is around 0.025Hz. The experiment starts from 16:40 and stops at around 16:40 the next day. The first finger touching happens at around 21:10. To have a double check, we re-touch the FPGA at around 21:50. The sampled counter values are shown in Fig. 7. The left zoom-in view shows the first finger touching. For the purpose of comparison, we show another zoom-in view on the right without finger touching.

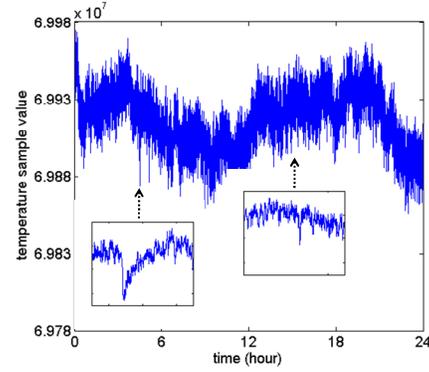


Figure 7: temperature samples within 24 hours

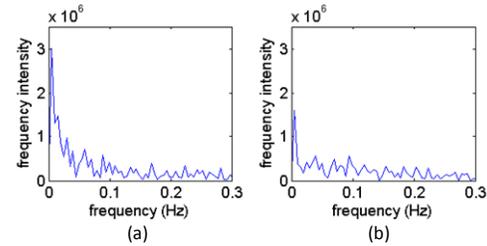


Figure 8: (a) Spectrum of the left zoom-in view; (b) Spectrum of the right zoom-in view.

Although it is clear that the large concave area in the left zoom-in graph corresponds to the finger touching, it is not easy to locate where the finger touching happens in the large figure without labeling it. We perform Fourier transform to get the spectrums of the two zoom-in areas. The results are given in Figure 8(a) and Figure 8(b) respectively.

In Fig. 8, it is obvious that the spectrum for the left zoom-in view has more frequency components between 0.005Hz to 0.05Hz which is caused by the finger touching. The post-process will try to keep these components and filter out others. The result of the post-process is shown in Fig. 9.

Fig. 9 presents the result of the overall experiment. We can clearly see two spikes corresponding to the finger touchings at around 21:10 and 21:50. The zoom-in views with and without finger touching are also shown. Again, we perform the Fourier transform on them and obtain Figure 10(a) and Figure 10(b).

Compared with Fig. 8, the unwanted high-frequency and low-frequency components are mostly filtered out in Fig. 10. That's why we can clearly see spikes for finger touching. By now, we have verified that our solution works correctly and reliably.

Our solution is not complex. It is low cost, and needs only an ordinary ring oscillator based sensor and an 8-bit micro-controller. Of course, the cost also depends on the SPE. If the major frequency of the SPE is high, then the sampling frequency should also be high, then the counter can be smaller. But at the same

time, the burden for the post-processing increases. Moreover, if the major frequency of SPE is close to other unwanted frequency components, more complicated post-processing should be implemented. This also requires a stronger processing capability.

Besides the stand-alone version, our solution can also be used as an IP module in a SoC system. In this case, the micro-controller can be replaced by the main processor, which reduces the hardware cost further. The costs of our design are as follows in Table 1.

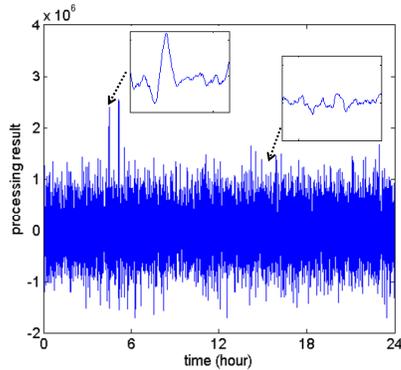


Figure 9: signal processing results

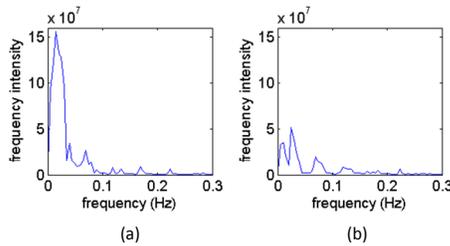


Figure 10: (a) Spectrum of the left zoom-in view; (b) Spectrum of the right zoom-in view.

Table 1: costs of the design

Integration choice	Slices	Block RAM
Stand-alone module	157	2
IP in an SoC	23	0

Furthermore, the division of the filter into counter based filter and software-based post-processing brings us the flexibility. For a system with limited hardware resources but a strong processor, we can choose an acceptable higher sampling frequency, which leads to more post-processing, to loosen the requirements for the hardware and vice versa.

In this experiment, a finger-touch as a SPE has no other special characteristics except for its unique major frequency component. Therefore, it is reasonable to expect the same result for other SPEs. Strictly speaking, our solution is generally applicable to monitor most of the low-frequency SPEs. When the SPE's frequency is higher than 100KHz, PicoBlaze is not capable to handle the post-processing.

6. Conclusion

In this paper, we present a solution to make a ring-oscillator based thermal sensor more sensitive to the thermal variation caused by a SPE. The method is based on a frequency filtering. We make use of the existing counter to build up a low-pass filter. Furthermore, we solve the intrinsic aliasing brought by the counter by increasing the sampling frequency. Experimental results based on FPGA demonstrate that, after the counter, simple low-pass and high-pass post-processing is enough to detect a person's finger touching on the FPGA package correctly and reliably. We conclude that our solution can be generalized to monitor other SPEs, especially the low-frequency ones.

References

- [1] H. Sanchez et al. *Thermal management system for high performance PowerPC microprocessors*. Digest of Papers – COMPCON – IEEE Computer Society International Conference, page 325, 1997.
- [2] B. Datta, *On-chip thermal sensing in deep sub-micron COMS*, Master thesis, University of Massachusetts, 2007
- [3] S. Velusamy et al, *Monitoring Temperature in FPGA based SoCs*. Proc. of International Conference on Computer Design (ICCD'05), page 634, 2005.
- [4] C. Marsh, T. Kean and D. McLaren, *Protecting designs with a passive thermal tag*, Proc. of Electronics, Circuits and Systems, ICECS'08, page 218, 2008.
- [5] G. M. Quenot, N. Paris and B. Zavidovique, *A temperature and voltage measurement cell for VLSI circuits*, Proc. Euro-ASIC 91, page 334, 1991.
- [6] S. Lopez-Buedo, J. Garrido and E. Boemo, *Thermal testing on reconfigurable computers*, Design & Test of Computers, IEEE, volume 17, page 84, 2000.
- [7] S. Lopez-Buedo, J. Garrido and E. Boemo, *Dynamically inserting, operating, and eliminating thermal sensors of FPGA-Based Systems*, IEEE Transactions on Components and Packaging Technologies, vol. 25, page 561, 2002.
- [8] A. V. Oppenheim, A. S. Willsky and S. H. Nawab, *Signals and Systems*, 2nd Edition, ISBN: 0-138-14757-4, Prentice Hall, 1997.
- [9] H. Nyquist, *Certain topics in telegraph transmission theory*, Trans. A.I.E.E, page 617, 1928. (See also Proc. IEEE, vol. 90, page 280, 2002)
- [10] C. E. Shannon, *Communication in the presence of noise*, Proc. IRE, vol. 37, page 10, 1949. (See also Proc. IEEE, vol. 86, page 447, 1998)
- [11] Digilent Inc. <http://www.digilentinc.com/>.