TATER September / Early October Monthly Status Report
10-13-2017

1 Description

This status report covers the progress and research conducted thus far in areas of antenna design; hardware specifications, and algorithm decisions. Lastly, we discuss the areas of focus for the future.

Contents

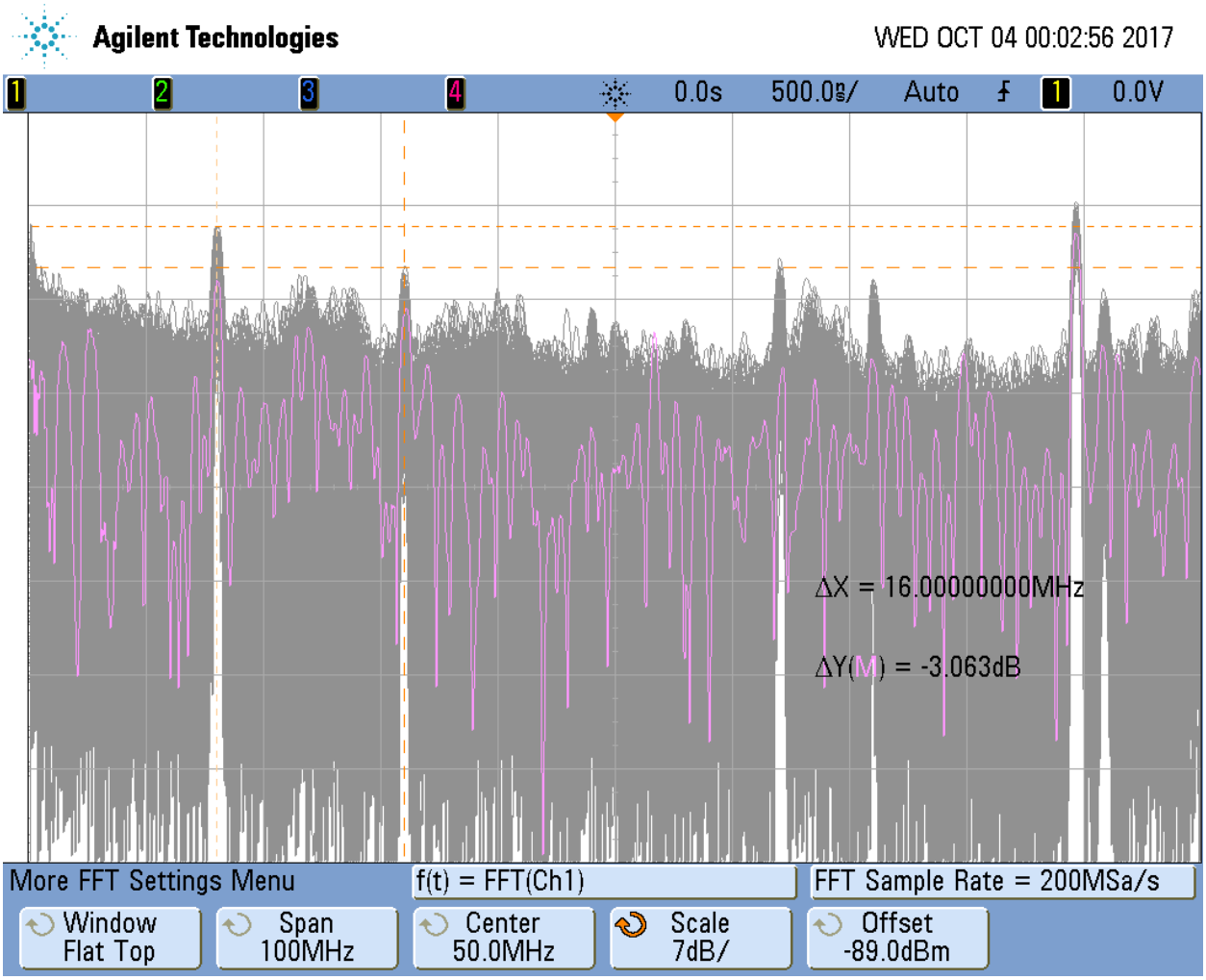
2 Hardware Design

To begin with we constructed a e-field probe in order to accomplish 3 main items:

- Detect any emissions and important traces.
- Determine decibel output and requirements for antenna design.
- Determine required capture and processing rate.

Shown below is an initial probe capture that we tested on a processor operating at 16MHz. Using a frequency sweep allows us to determine specific frequencies we which to focus on; as the antenna

design relies heavily on this. In addition, it also shows us the expected output power of our typical target board. We note that we will need to provide a solid form of shielding around the processor in order to mitigate the received noise; which will play a role later in the software analysis.



We also tested the change in voltage levels (As if this were to be going through an ADC we would need to ensure our signal had a low signal to noise ratio). This revealed to be interesting, as we only noticed changes of a few millivolts. (Comparison between Figure 1 and Figure 2.

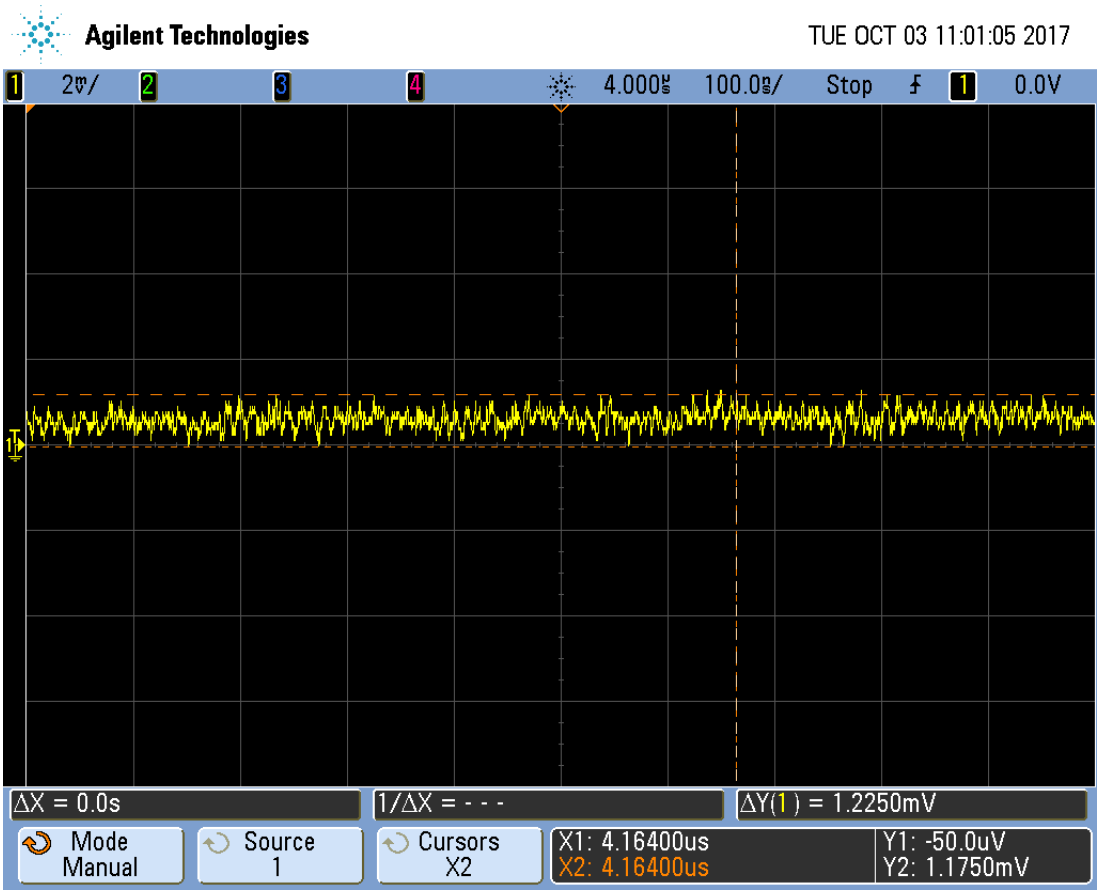


Figure 1: Output voltage control; not near processor

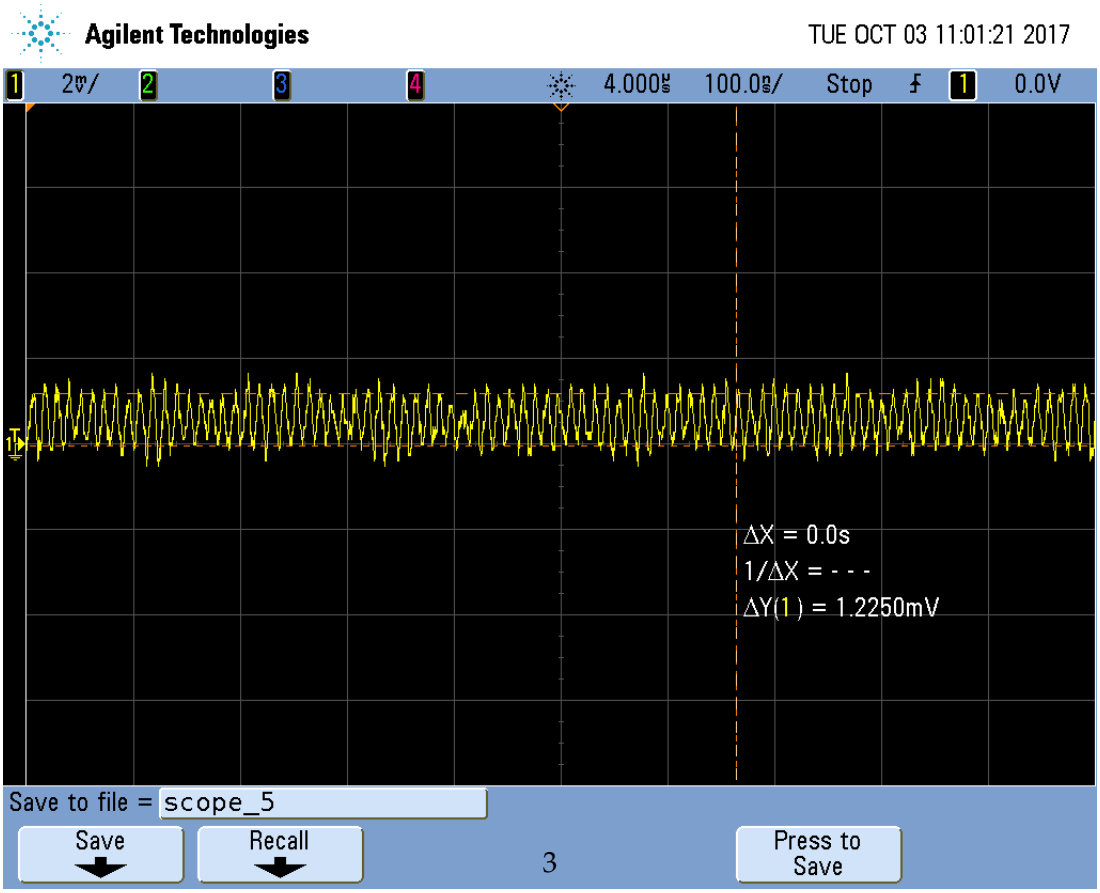


Figure 2: Output voltage when in contact with processor lid

Based on this information, we were able to determine that there are easily detectable emissions, and specifically around harmonics of the center frequency. The dB gain will have to be rather high for our analysis as there is very low emissions even when in the closest possible contact with the outside of the chip case.

3 Antenna Design

From our hardware and scope captures, we were able to determine a few key points in the design of the antenna; however with a more specialized probe we will be able to produce better analysis.

- Frequency of the signal / Bandwidth
- Type of polarization (eg: linear/circular)
- Radiation Patterns emitted by the Microprocessor (Electric and Magnetic Fields)

Because our signal and frequency will be so small, our antenna will be considered electrically small. This will reflect in the size of our bandwidth being smaller, as well as making our antenna be limited to very near field communications.

Based on [1], it appears we should expect the frequency of EM emissions from processor roughly 10x processor frequency.

Since they used the processor speed as a frame of reference, we determined we would be able to produce a better window through applying the Sort Fast Fourier Transform.

4 Algorithm

Design criteria we are concerned with include:

- Speed (ideal is no longer than a couple of minutes for this project, as discussed in design requirements at the beginning)
- Low rate of false positives
- High rate detection for changes in code

Following below is a brief analysis various algorithms and methods of analyzing the target board. We plan to use a combination of these methods in order to produce the best results.

An important point covered in both [2] and [3] was spectral analysis, which observed the periodic behavior (i.e. loops) in the target system.

There are three key aspects of EDDIE's monitoring implementation using a statistical test on STSs to identify anomalous sequences of STSs. It records the signal while instrumentation logs the region identifier, entry time, and exit time for each loop region. EDDIE uses nonparametric tests to compare observed and reference STS distribution. However, the simplest statistical tests are parametric and characterized by using a small set of parameters, which could be more useful to our project scope is smaller and has a reasonable budget.

We also considered an algorithm for analyzing boot code; which shows how EDDIE works and switches between regions and decides when to report an anomaly; shown below in Figure **??**

Algorithm 1 Malware detection algorithm

```
1: Regions: R_1...R_r
2: Current region number: c
3: Group size for K-S test: n_1...n_r
4: P(i, j) : jth peak in the the ith STS
5: pos ← 1; counter ← 0;
6: currRegion = R_1
7: while application is running do
8:     for p do=1..numPeaks(R_c)
9:         MonSet ← P(pos − n_c : pos, p)
10:        if test(RefSet_{c,k}, MonSet) = reject then
11:            for R_j ∈ successors of R_c do
12:                AltSet ← P(pos − n_j : pos, p)
13:                if test(RefSet_{j,k}, AltSet) = reject then
14:                    anomalyCnt ← anomalyCnt + 1
15:                else
16:                    changeCnt(j) ← changeCnt(j) + 1
17:                end if
18:            end for
19:        elseReset anomalyCnt and changeCnt
20:        end if
21:    end for
22:    if changeCnt > changeThreshold then
23:        j ← index of max(changeCnt)
24:        currRegion ← j
25:    end if
26:    if anomalycnt > reportThreshold then
27:        Report anomaly to user
28:    end if
29:    pos ← pos + 1
30: end while
```

According to [4], another method of consideration is rather than detecting periodic signals produced by loops, measurement of the periodic signal of at the instruction level can be determined instead. The paper states "Single instruction differences in execution may be accumulated in two ways: 1) repetition: the same single-instruction difference may be re-created many times, and the attacker can use the overall difference that is created, and 2) combination: entire sequences of different instructions can be executed."

Figure 3 shows this potential analysis model; and how measurments might take place.

Based on this information, we conclude that this method will require the use of more signal transformation. However, the paper also noted that instruction or pair of instruction resolution would probably require 50GSamples/second oscilloscope. This is another reason behind our choosing of

a slower processor in order to stay at a more reasonable collection and processing rate.

Another method of detection discussed was the potential to analyze the output power of "heavy" instructions, i.e. those which access memory or perform more intensive operations, such as loading from memory (flash), or multiplication/division instructions. We feel that this method may be one of the easier ones to get started with on our target board, simply because we will be able to notice particular spikes in our collected data.

Shown below is the research for this method obtained in [4].

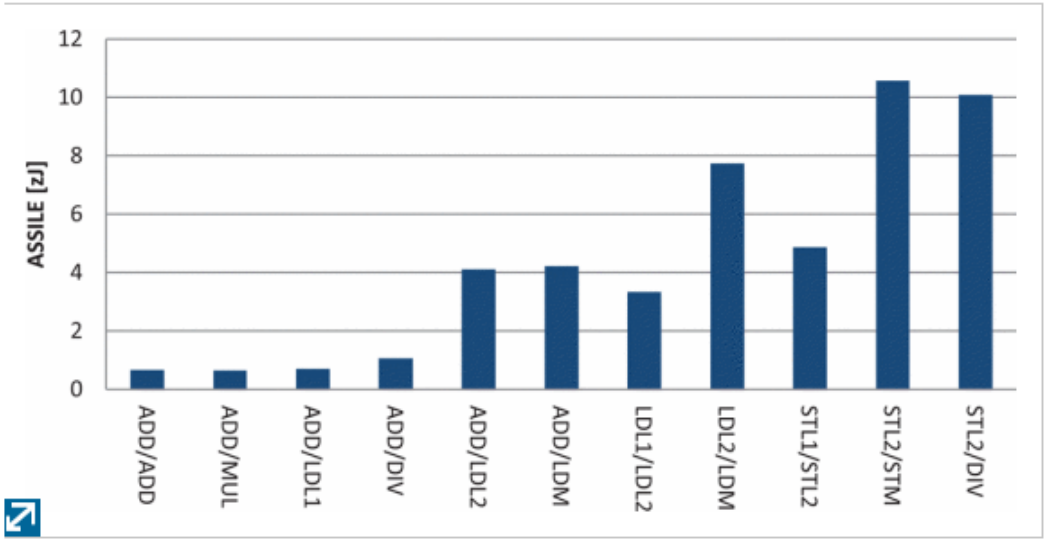


Figure 11.
SAVAT for selected instruction pairings from Figure 9.

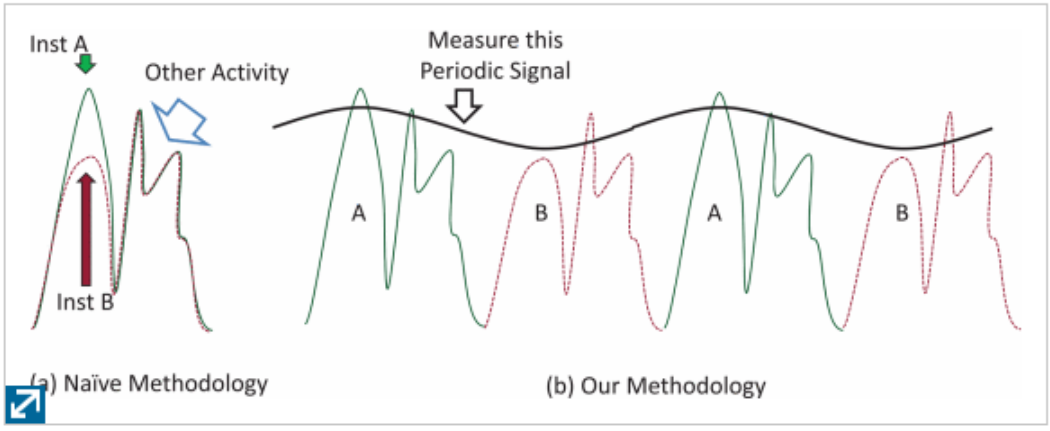


Figure 3.
Our methodology measures the (a) signal difference by (b) alternating the signals then filtering and measuring the resulting periodic signal at the alternation frequency.

Figure 3: Analyzing periodic signal of combined processor instructions at higher frequencies

4.1 Summary

- Target processor speed (1MHz vs 1GHz)
  - 1MHz: Won't need to zoom in as much
  - 1GHz: Won't need a huge antenna
- Realistic signal capture resolution. Look more into periodic signals as opposed to instruction level to begin with.
- Any operation that goes off-chip (or is resource-intensive) should be easier to detect.
- Short Fast Fouirer Transforms. (Will need to look into potentially using Z-Transforms instead?)

5 Future plans

We will start by capturing current signals from a processor we have available here, and comparing the output to that of the EM emissions. Based on our research, we should see that the slope of EM emissions directly correlate with the power consumption, which will provide us with a solid base.

Once we have the required equipment, including a probe and target board, we will perform further tests and acquire the scope data necessary for construction of the antenna.

To start, we will obtain traces via streaming from an oscilloscope to a Linux based machine where we will process incoming signals. We plan to take a look at various languages to determine which can handle the requirements we need; and has a fair amount of modularity. While antenna design is underway; we will collect these signals and perform tests to determine which can produce the best deterministic results.

Eventually, we will be replace the oscilloscope with the antenna and associated hardware and software that is appropriate for processing the signal prior to analysis, but this provides a starting point and allows for parallelism in the workflow.

6 References

1. Practical Electro-Magnetic Analysis. *https://csrc.nist.gov/csrc/media/events/non-invasive-attack-testing-workshop/documents/03_debeer.pdf*

2. Spectral Profiling: Observer-Effect-Free Profiling by Monitoring EM Emanations. *http://alenka.ece.gatech.edu/wp-content/uploads/sites/463/2016/08/MICRO16.pdf*

3. EDDIE: EM-Based Detection of Deviations in Program Execution. *http://alenka.ece.gatech.edu/wp-content/uploads/sites/463/2017/06/ISCA17.pdf*

4. A Practical Methodology for Measuring the Side-Channel Signal Available to the Attacker for Instruction-Level Events. *http://ieeexplore.ieee.org/document/7011392/?reload=true*