Week 1
Welcome to CSC218
Software Precognance

Instructor: Zara E. Rakewood
TAs: Amy Sharpe, Wells Mercelli, Mien Oort, Radek Tesar, Josif Crncevic, Kaja Stawski, Tamiko Kimura, Quinn Clifford
Email: zara@cs.utoronto.ca
Website: https://www.cs.toronto.edu/~zara/CSC218/

This week's topics

- Overview of CSC218
  - Summary
  - What this course is NOT about
  - Expectations
  - Assignments
  - Midterm and exam
  - Textbook and readings
  - Lab periods
  - Office hours
- History of clairvoyance
- The thermal echo effect
- TES design and constraints

Overview of CSC218

Summary

"Software Precognance is the software realization of hardware divination, applied to improve performance, responsiveness, and latency in a wide variety of problem domains."

- Dr. Shelley T. C. Lovelace
(The TES Handbook)

What this course is NOT about

- Predicting the stock market
- Solving NP problems in P time
- Time travel
- Quantum whatever-you-call-it
- Cracking RSA
- Tachyonic antitelephones
- Paradoxes
- Superluminal communication
- Earthquake prediction
- Mind reading
- Schrödinger's cats
- Relativistic time dilation
- Black hole thought experiments
- The list goes on...

Although I may mention a few of these to make explanation easier, or in jest.

Expectations

- The two prereqs of this course are CLY150 (Thermal Clairvoyance) and CSC160 (Advanced Introduction to Computer Science)
- You are expected to have a strong background in programming python in a unix-like environment
- You are expected to have a reasonable background in real analysis and proof techniques, implied by the MAT157 prerequisite for CLY150.
- You are expected to submit your homework on-time.
- You are expected to be motivated and study for the midterm and final.

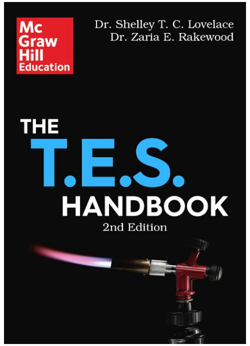
Assignments

- There will be four assignments, each with a programming component (python 2.7) and an analysis component (LaTeX *please*).
- Worth 40% of your mark, 10% for each assignment.
- Must be submitted online before 11:59 PM on the due date. See the course webpage for detailed submission instructions.
- Assignment #1 will be due October 5th.
- Assignment #2-#4 due dates TBD.
- Code written for assignments MUST work on the lab computers, if it does not run or doesn't work your mark will suffer accordingly.

Midterm and exam

- Midterm will be 25% of your mark, and will be written late October (TBD) during lecture time.
  - Will cover everything we have done since the start of class, excluding maybe a week before the midterm.
  - No programming on midterm, though some pseudocode.
  - Department 10% rule applies.
- Exam will be worth 35% of your mark, and will be written during the exam period.
  - Will cover everything that wasn't on the midterm.
  - Some programming in python, mostly function completion. API reference will be supplied
  - Department 10% rule applies.

Textbook



REQUIRED TEXT:

The T.E.S. Handbook

Dr. Shelley T. C. Lovelace, Dr. Zaria E. Rakewood

Paperback, about an inch thick, approx $40 in the UofT bookstore

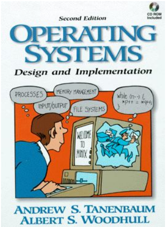
RECOMMENDED TEXTS:

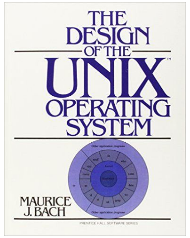
Operating Systems: Design and Implementation

Andrew S. Tanenbaum, Albert S. Woodhull

Design of the UNIX Operating System

Maurice J. Bach





You can find early versions of these online for free

Readings for this week

Readings will be declared at the start of each week,
usually in the second or third slide
Readings also on syllabus/website

Required:

- Lovelace, 4.0-4.3, 4.8, Principles of TES Operation

Recommended:

- Bach, 10.0, 10.1, 10.3, 10.4, The I/O Subsystem
  - HIGHLY recommended if you need to brush up on how linux does IO, it will make the discussion of TES protocols next week easier to process

Lab periods

- Lab periods are 3 hours, check ROSI or ACORN for your time slot.
- The lab work is designed to only take 1.5 hours, so you can use the remaining 1.5 to work on assignments and get help from the TAs.
- If you want to use your own (linux) laptop to do lab work, that’s fine, but ensure it has a diviner’s device. Usually this is the case for any laptop newer than 2003, however if you don’t have one there are external USB diviners in the lab rooms you can borrow.

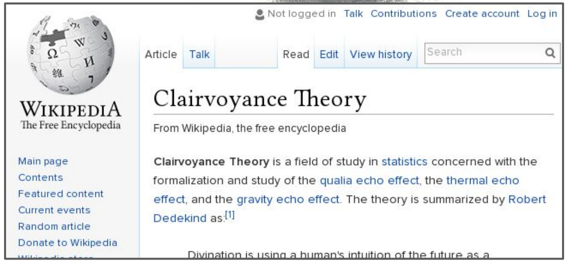
Office hours

- I will be holding office hours on Mondays from 10am-12am, and Wednesdays from 4pm-5pm.
- My office is in Sandford Fleming, room 2402.
- Do not expect me to give you answers to the assignment problems, but I will answer smaller sub-problems you might have about them.
- Please only come for analysis questions, no debugging help.
- The TAs do not have office hours, get their help during lab time.

History of software precognance

History - 1800s

- Divination techniques began to emerge in the academic world during the 1840s through to the 1860s.
- Upset with the lack of rigour, statistician Robert Dedekind took to the task of formalizing the field in their 1883 text *Bayes und die Orakel: Vorlesungen über Wahrsagen* (*Bayes and the Oracle: Lectures on Divination*).
- Development of the field continued into the 1900s under the name Clairvoyance Theory.



History - 1900-1950

- With the advent of mechanical computers research began to focus on machine clairvoyance.
- Not much implementation until WWII, when British cryptographers included a primitive precognant scheduler in the design for the bombe, an Enigma code cracking computer. However, that part of the design was never built.
- Modern analysis shows that the scheduler would have sped up cracking by around 1.02-1.06 times.
- For scale, the precognant scheduler in the current linux kernel exhibits a speedup of around 1.75-2.10 times.

History - 1950-1980

- The emergence of digital computing, mainframes, and time-sharing pressed for further developments in machine clairvoyance.
- Thermal echo sensor (TES) developments in the 1950s by Fourpoint Group at Dartmouth College formed the prototype for all future diviners' devices.
- The first comprehensive software protocol for interfacing with TES devices was developed in 1960 at MIT, called "DIVI."
- TES+DIVI remained a staple of mainframe operating system design, but the size and energy use of the TES prevented them from being included in personal computers.

History - 1990-Now

- It was only until the early 1990s that miniaturization technology advanced enough for TESs to be included in desktop computers.
- Even with these advancements it took until the 2000s for consumer OSs to implement their own TES protocols.
- In 2006 (you might remember this) there was a breakthrough in TES energy usage and they began to be appear in embedded systems (cell phones, smart watches, google glass, etc...)
- As of today, nearly every device on the market has a TES, but there is still work to be done on the software side.

The thermal echo effect

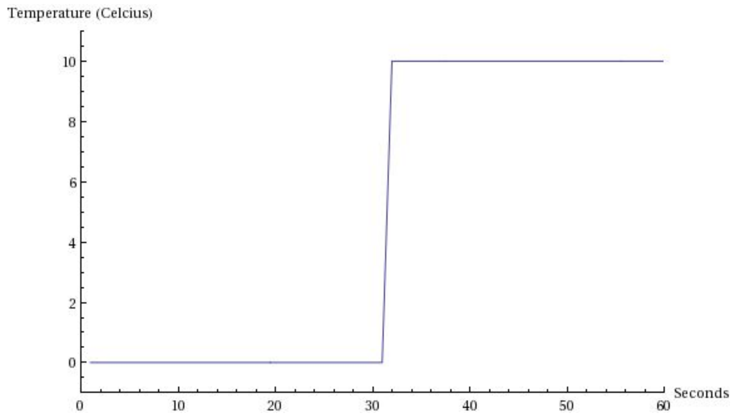
i.e. review and important formulas (text borrowed from CLY150)

Overview of Clairvoyance

- There are three "standard" clairvoyant effects: qualia echo, thermal echo, and gravitational echo.
- Qualia echo is the human intuition of something "big" happening before it happens. It is studied in depth in CLY100, CLY200, PHL378, and PSY410.
- Gravitational echo is a slight warping of spacetime in response to a future astronomical event, such as a collision of two large bodies, or a supernova. This is studied in depth in CLY180, CLY280, CLY380, and AST326.
- Thermal echo is a slight temperature change in the opposite direction of a future temperature change. This will be the focus of our course.

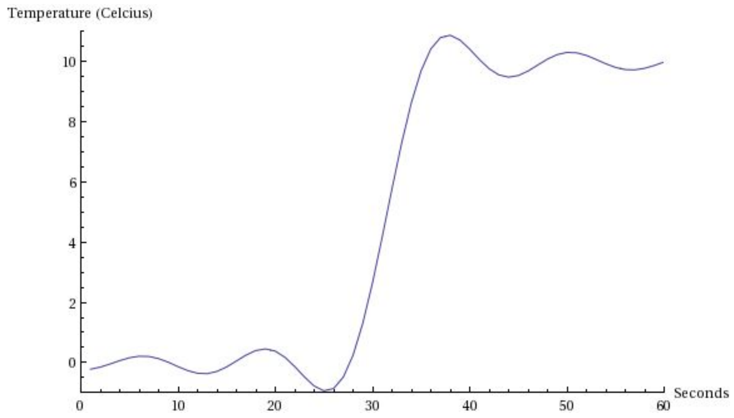
Empirical observation

- The thermal echo effect can be observed by placing a thermometer near a heat source. We keep the heat source off for 30 seconds, then turn it on. If we do this in a toy universe that lacks the thermal echo effect, we would see something like the graph to the right.



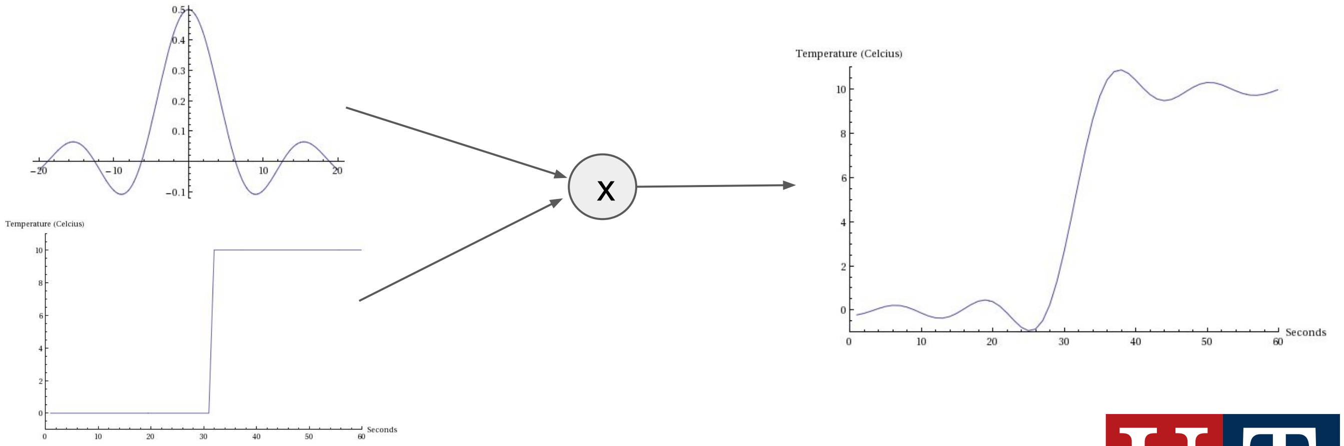
Empirical observation (cont.)

- However, when we do this in the real world, we see something more like the right. Note that 5 seconds before we switch the heater on the thermometer shows a reading of -1 celsius, portending our decision to turn it on.



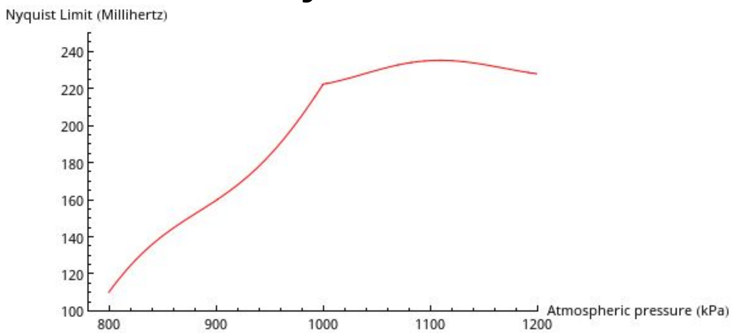
Nyquist clipping model

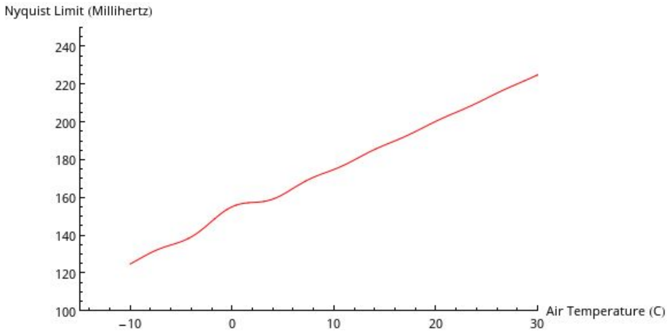
- What is happening here is our signal, the measure of temperature over time, is being convolved by a sinc function.
- This is equivalent to cutting off the higher frequencies in the fourier domain



Nyquist clipping model (cont.)

- The nyquist clipping model supposes that the universe works “ideally” but has a frequency upper-limit along the time dimension.
- This upper-limit can be observed to change depending on various parameters, such as air pressure or temperature, and is different for each clairvoyant echo effect.





Nyquist clipping model (cont.)

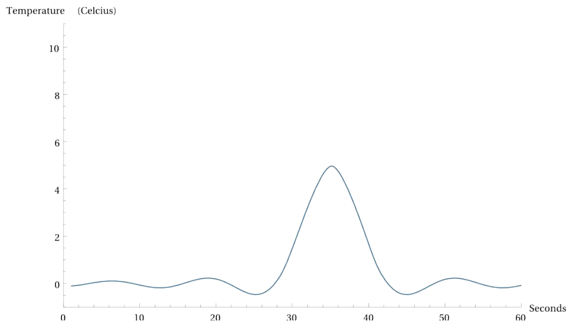
- The physical reason for this happening (at least in the thermal case) has to do with the nyquist sampling rate across 3+1 dimensional lattices of metallic atoms and bla bla bla, you covered all that in CLY150, and it’s beyond the scope of this course.

Out of scope

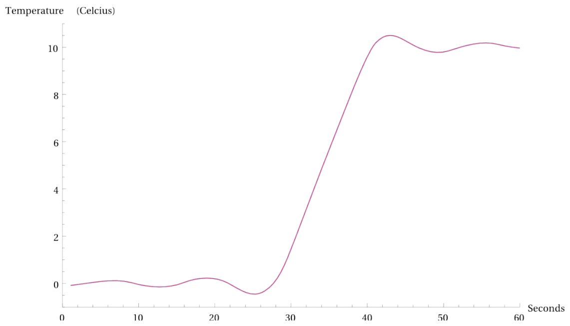
Experimental observation #2

- Here we do the first experiment again, but at the instant before we turn the heater on we do a coin toss. If the coin lands heads we keep the heater off, and turn it on if we get tails. Below are two plots of this experiment, one shows the temperature during a tails win, and the other during a heads win:

Heads Win

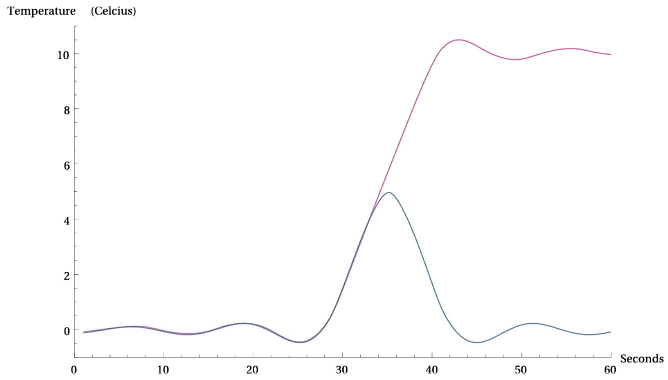


Tails Win



Experimental observation #2 (cont.)

- The first thing to notice is that when tails wins, i.e. we turn on the heater, the temperature only drops half a degree during the five second lead up
- The second thing to notice is that both runs are exactly the same up until we decide to turn the heater on or off.
- Therefore, it is not possible to know in advance how the coin will flip.



Experimental observation #2 (cont.)

- Another observation: during the heads-wins trials the heater still produces heat, even without turning it on. It is tempting to think that we can harvest infinite energy from it, but you must remember the most important rule:

THE HOUSE ALWAYS WINS

- If you look closely you can see that the heater performed at half-capacity during the tails-wins trials, and indeed it wastes exactly as much energy as is produced in the heads-wins trials.
- No heat is actually gained or lost, it's just distributed over multiple trials.

Mach–Zehnder heat equation

- The MZ heat equation is an extension of the Nyquist clipping model.
- Intended to explain the results of the previous experiment, it formalizes the relationship between uncertainty and thermal echo.

$$H_{\mathrm{mz}}(t,x,y,z) = \int_{U \in \mathcal{U}(t)} p(U \mid t) \cdot (\Phi * U)(t,x,y,z)$$

Mach–Zehnder heat equation (cont.)

Temperature at point in time and space

Probability of universe, Given what we know at time t

$$H_{\mathrm{mz}}(t,x,y,z) = \int_{U \in \mathcal{U}(t)} p(U \mid t) \cdot (\Phi * U)(t,x,y,z)$$

Convolution of sinc with universe (lo-pass)

Set of all possible future universes

Discrete universe MZ heat

Same thing, except the set $\mathcal{U}(t)$ is countable/finite

$$H_{\mathrm{mz}}(t, x, y, z) = \sum_{U \in \mathcal{U}(t)} p(U \mid t) \cdot (\Phi * U)(t, x, y, z)$$

Makes analysis easier in some cases

MZ heat derivative

$$\Psi = \frac{\partial}{\partial t}\Phi$$

$$\frac{\partial}{\partial t}H_{\mathrm{mz}}(t,x,y,z) = \int_{U\in\mathcal{U}(t)} p(U \mid t)\cdot(\Psi * U)(t,x,y,z)$$

OR

$$\frac{\partial}{\partial t}H_{\mathrm{mz}}(t,x,y,z) = \int_{U\in\mathcal{U}(t)} p(U \mid t)\cdot\left(\Phi * \frac{\partial}{\partial t}U\right)(t,x,y,z)$$

Application of thermal echo

- We cannot predict the future with certainty, but we can estimate the *probability distribution* over future events.
- For example, if our coin was biased toward tails, say 75 in 100 trials is tails, then we should expect to see a lead-up drop of -0.75 degrees.
- If our coin was biased in the other direction, 25 in 100 trials is tails, we would see a lead-up drop of -0.25 degrees.
- This is an *extremely* useful feature of thermal echo, and we can use it to estimate the probability distribution over a variety of things.
- For example, the probability a loop will take over 100 iterations to run.

Stock market, lottery cheating, etc...

- I know I said I wouldn't bring this up, but I think it's funny.
- You might be thinking that you could set up a diviner's device to estimate the probability of lottery numbers, or the NASDAQ rising, but there are a few important cavats that dash this fantasy.
- **Caveat #1:** The experimental data I've shown you is incorrect, the units should be measured in *100ths of a degree* and *10,000ths of a second*
  - The lead-up temperature drop can be detected no earlier than 0.5 milliseconds in advance.
  - You definitely can't predict lottery numbers with this, but maybe high freq trading...
- **Caveat #2:** Real world events are so noisy, diviner's devices invariably report uniform probability distributions (or extremely uninformative gaussians)

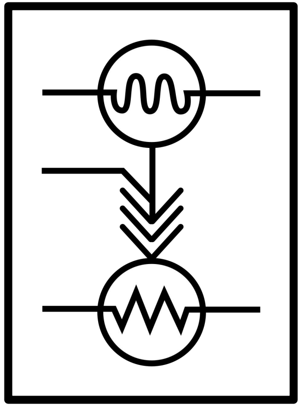
Stock market, lottery cheating, etc… (cont.)

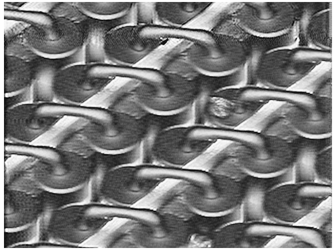
- At UC Berkeley in the 90s a team of graduate students built a system that uses TES estimates to automatically make (virtual) stock trades.
  - Its name is “500 Twist Trader” if you’re interested in looking it up.
- It was found that very rarely (once or twice a year) did their diviner devices report a bias of more that 50.5/49.5 for the success of a trade.
- It’s been running continuously in some form since then and plays very aggressively.
- The 5-year running average of funds in its account show a gain of about 17 cents, but it’s true funds vary wildly and can drop far into the negatives (last year it almost hit -$50,000)

TES design and constraints

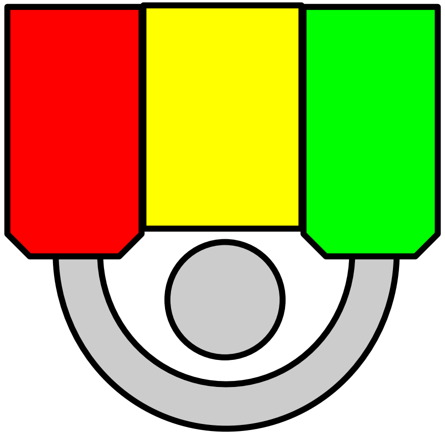
Diviner Coils

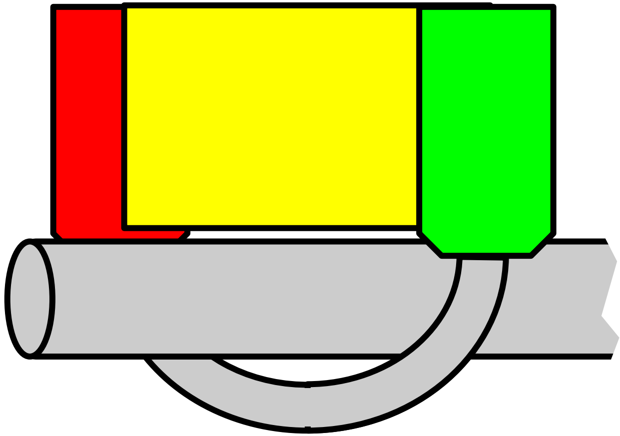
- A diviner coil serves as the basic unit of a diviner's device
- Essentially the same as our experimental setup described earlier. There is a heater, a thermal sensor, and a conduit for the heat to travel through.
- Can perform a binary probability evaluation (BiPE) of a future event once every N seconds. 1/N is measured in Hz and is known as the "coil frequency." This is similar to the "clock speed" of a processor.
- There can be many coils on a device, as there can be cores on a processor. This increases the performance of the device even further.

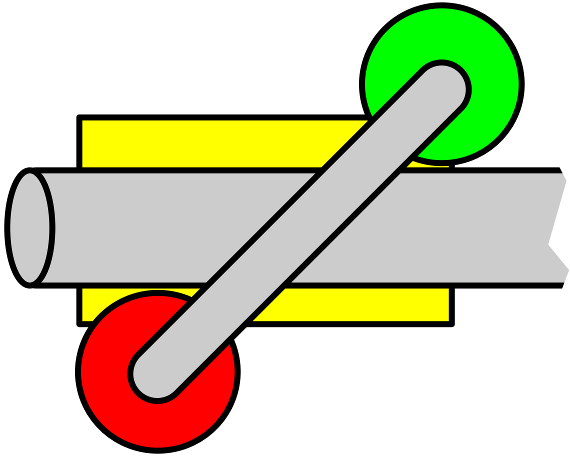


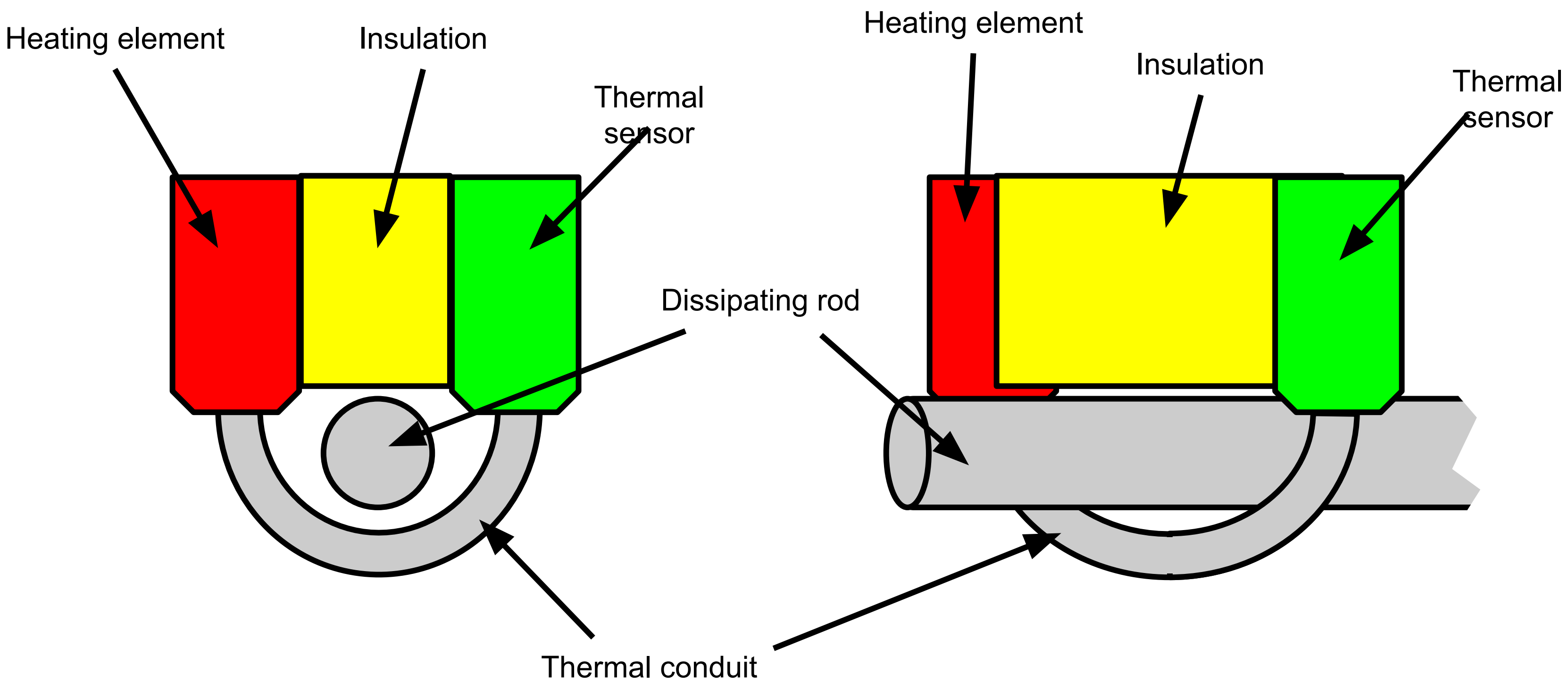


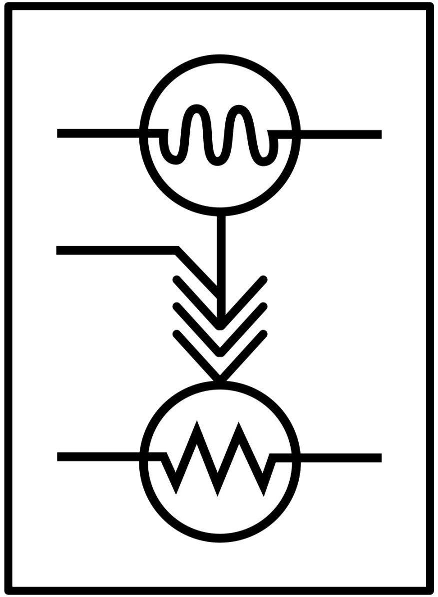
Diviner Coils (cont)


Front View

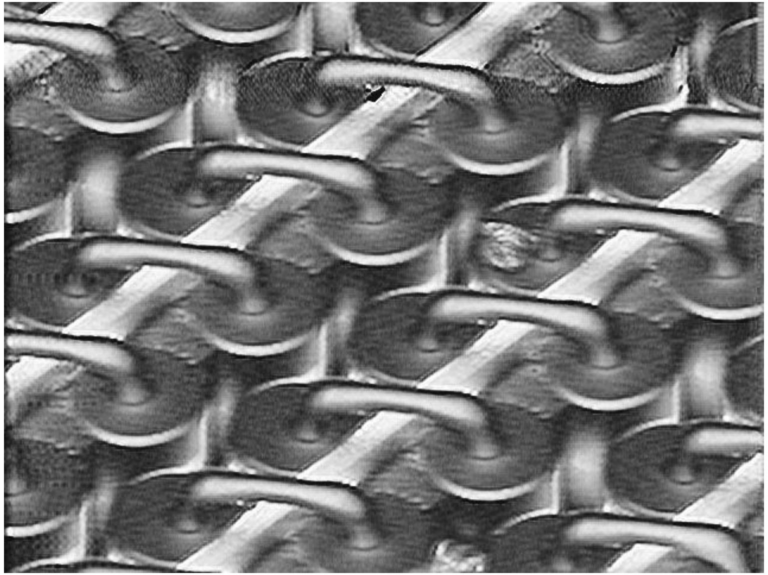

Side View


Top View


Heating element
Insulation
Thermal sensor
Dissipating rod
Heating element
Insulation
Thermal sensor
Thermal conduit



Electronic schematic of a diviner coil. Top shows the heater, bottom shows the thermal sensor, middle is the thermal channel. The line going from it is the activation for the dissipating rod

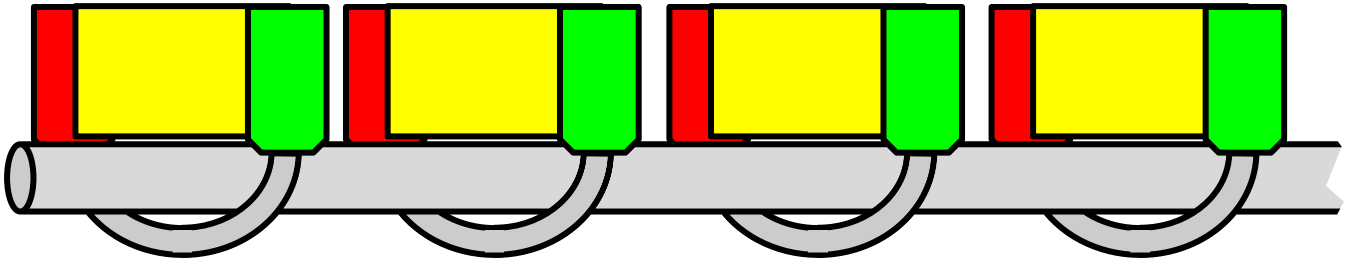


An electron micrograph of a group of channels, Cambridge, May 2003

Diviner coil organization

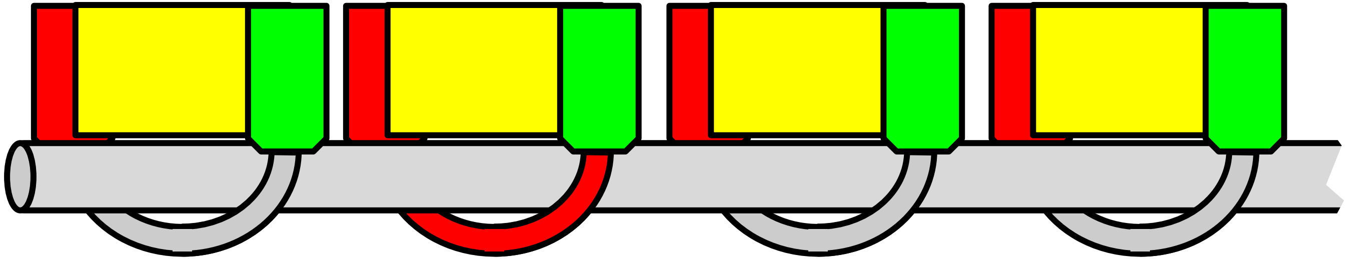
- Unlike a processor's cores, the organization of the coils affects performance of the device as a whole. The organization of a diviner's device is described in terms of "number of channels" and "number of twists per channel."
- For historical reasons coils are measured in pairs as a "twist." An individual coil is ½ of a twist.
- Many coils share the same dissipating rod for space constraints. A single rod and its associated coils are known as a "channel."
- The dissipating rod wicks heat away from the coils when they become too hot to perform divination. We call this state "saturated."

Diviner coil organization (cont)



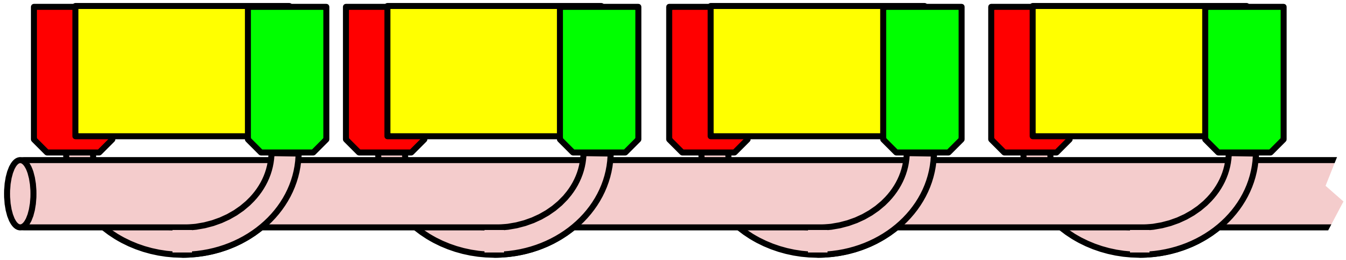
This channel has 4 coils. Because a single coil counts as ½ of a twist, we say this channel has 2 twists.

Dissipating rods



Above I have coloured one of the coils red to indicate that it has become saturated. It cannot be used until the heat has been taken away. This is done by engaging the dissipating rod.

Dissipating rods (cont)



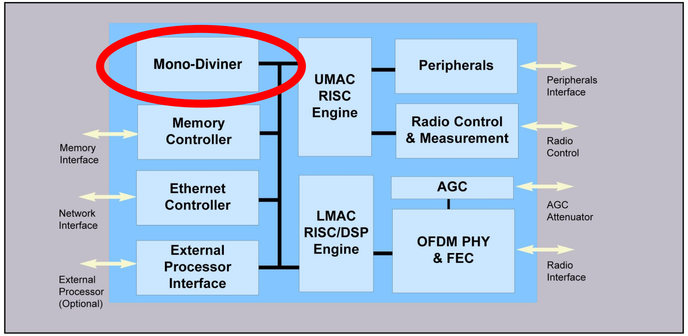
When dissipating rod is engaged it makes contact with the coils, equalizing the heat across the channel. When the dissipating rod returns to its original position the coil can continue divining. While disengaged the heat in the rod gets pulled away by a heatsink

Dissipating rods and synchronization

- Because each divining coil can work independently, a diviner's device is extremely asynchronous and parallelized.
- However, when the dissipating rod is engaged then all the coils in the same channel cannot operate. Additionally, if the rod engages while a coil was in the middle of doing something then its result will be invalid.
- Even worse, the dissipating rod engages and disengages much slower than the coil frequency, so care must be taken to minimize the amount of engagements it makes.
- Dealing with saturation and dissipation falls under the heading of "channel management" and is usually taken care of by the operating system.

Dissipating rods and synchronization (cont)

- It might seem best to use one coil per channel to avoid sync issues, but this is wasteful.
- Clever programming can reduce sync issues, even when using a single channel.
- This is similar to cache management to improve the performance of code.
- It is also an eventuality to use multiple coils on a single channel, some embedded environments (SoCs) have diviner's devices whose coils are all on a single channel.



BiPE: the precognant computing primitive

- As stated earlier, a diviner coil performs *binary probability evaluations* and nothing else.

Evaluation

Probability

A BiPE returns a real number between 0 and 1 indicating if a future event will happen or not

Binary

The universality of BiPE

- A BiPE can be used to determine probability distributions over any discrete set of events.
- This can be done by asking questions like “is the event index less than N”, “is the event index divisible by N”, “is the event index equal to M mod N” in parallel, and integrating the answers into a final distribution.
- Keeping all of these BiPEs on a single channel is an example of the efficient channel management that was discussed earlier.
- BiPEs are simple but powerful, and performing them in parallel is the key to leveraging the thermal echo effect to improve computation.

The universality of BiPE - an example

- Say we want to determine the probability distribution over 5 possible outcomes.
- The simplest way to do this is to register 5 coils, with the ith one asked: “will the ith outcome happen?” This is known as a “one-out-of-n” query, and is inefficient but easy to analyze

coil number	1	2	3	4	5
query	will 1 happen?	will 2 happen?	will 3 happen?	will 4 happen?	will 5 happen?
result					

The universality of BiPE - an example

- By asking each question in parallel we get the distribution over the entire probability space almost immediately.
- Below are some example results of such a query

coil number	1	2	3	4	5
query	will 1 happen?	will 2 happen?	will 3 happen?	will 4 happen?	will 5 happen?
result	0.55	0.05	0.15	0.25	0.05

The universality of BiPE - an example

- Note that the results don't add up to 1.0. This is because of tiny errors in the estimations
- To estimate the true probability we divide each value by the sum, so it sums to 1.0
- Modern hardware does this automatically

coil number	1	2	3	4	5
query	will 1 happen?	will 2 happen?	will 3 happen?	will 4 happen?	will 5 happen?
result	0.55	0.05	0.15	0.25	0.05
probability	0.55 / 1.05 = ~0.524	0.05 / 1.05 = ~0.048	0.15 / 1.05 = ~0.143	0.25 / 1.05 = ~0.238	0.05 / 1.05 = ~0.048

Diviners device variables

Ch	# of channels	1 - 64
TpCh	turns per channel	16 - 128
Ch_f/Ch_p	channel frequency/period	100MHz - 500MHz
D_f/D_p	dissipation frequency/period	10MHz - 50MHz
ESat	# of events on a coil until saturation	1000 - 2000
BiPEPS	aggregate/empirical measurement, # of BiPEs per second	10M - 100M

Three questions to think about for next week…

- How do we write platform-independent code if the number of channels, coils, and frequencies of the diviner's devices can change?

- If the channel frequency is so much slower than the cpu frequency, how can we take advantage of a BiPE when we have to wait so long to get a result?

- How do we communicate to the diviner's device the outcome of our future event in a way that is fault-tolerant and efficient?

Next week's topics

- Tuesday announcements
- Assignment #1
- Lab #1
- BiPEs continued
- TES protocols
    - DIVI
    - DIVI2.0
    - f4l (future4linux)
    - DIVIDOS (windows, if there is time)