Robustness in depth:
Rebalancing efficiency and reliability in the computational stack

Dave Ackley
UNM Computer Science Colloquium
December 4, 2009
Conclusions

- Robustness NAND Efficiency
- Design pressures
  Open horizons
  Fungibility
  Spatiotemporal stationarity
  Default active
  Check your own work
- Demos: Sync, State, Energy
1931
1931
- Unemployment 15.9%
1931

- Unemployment 15.9%
- Electric shaver
1931

- Unemployment 15.9%
- Electric shaver
- TV camera patent filed
1931
- Unemployment 15.9%
- Electric shaver
- TV camera patent filed
- Godel's incompleteness

Über formal unentscheidbare Sätze der *Principia Mathematica* und verwandter Systeme I
1931
- Unemployment 15.9%
- Electric shaver
- TV camera patent filed
- Godel's incompleteness

Dec 4, 1931: Movie premiere in NYC
The Bush Differential Analyzer
Integrator, Torque Amplifier
Analog: One Man's Signal..

Ideal linear amplifier ->
No information loss
Digital: Is Another Man's Noise

Ideal linear amplifier -> No information loss

Ideal non-linear amplifier -> Massive information loss
Going Digital: Massive Redundancy

Ideal linear amplifier -> No information loss

Ideal non-linear amplifier -> Massive information loss

An entire amp for every single bit!
1944: Harvard Mark I
IBM Automatic Sequence Controlled Calculator
- Digital, relays and motors
- 3 IPS
- Imperative: Cycles are precious!
The Deal
How? Remove redundancy

Be Efficient
Software Logical
Hardware Physical

Be Reliable
How? Add redundancy
And For Decades It Was Good
Except that Software Sucks
- Oh yeah, error handling
- Object oriented fantasies
And For Decades It Was Good Except that Software Sucks
- Oh yeah, error handling
- Object oriented fantasies
- Bad Old Days: Buffer overflows, wild pointers...
- Bad New Days: Thread hazards, mystery libraries..
And for decades it was good
Except hardware reneged
- CPU speed
And For Decades It Was Good
Except Hardware Reneged
- CPU speed
- Parallel, but how?
- Networks of independent components are inherently unreliable
Robustness in depth

Open horizons
Design for unbounded parallelism; node cannot hold whole

Fungibility
Common codebase; 'cell' differentiation based on context

Spatiotemporal stationarity
Avoid space-time privileged points, e.g., 'root node', 'setup phase'

Default active
Be useful out of the box; reflexes + 'conscious control'

Check your own work
Declarative AND imperative; deploy the test suites
The Illuminato X Machina Demos

- Sync
- State
- Energy
Sync

Blinky sync

```c
void seeFlash(u8 *) { // runs when we receive an 'f'
    if (digitalRead(ledPin) == LOW) // if we're already on when they flash
        ++wait; // wait a little longer
}

void setup() {
    pinMode(ledPin, OUTPUT); // sets the digital pin as output
    Body.reflex('f', seeFlash); // create a reflex triggered by 'f'
}

void loop() {
    wait = 500; // assume we'll wait a half second
    digitalWrite(ledPin, LOW); // sets the LED on
    println("f"); // send f packet to all neighbors
    delay(wait); // waits for a half second
    digitalWrite(ledPin, HIGH); // sets the LED off
    delay(wait); // waits for a half second

    while (buttonDown()) // as an extra feature
        delay(1); // push the button to add delay
}
```
Sync
Better sync
Better sync yet
State

- But what about making decisions?

Scalable TTT
Energy
- Being embodied.
Power sharing
It's time for a new deal

Robustness in depth

Software must own robustness
- Leveraging assumptions always costs
- Make sure they're worth it

Hardware must own awareness
- Reflexes right or wrong
- The dominance of the real