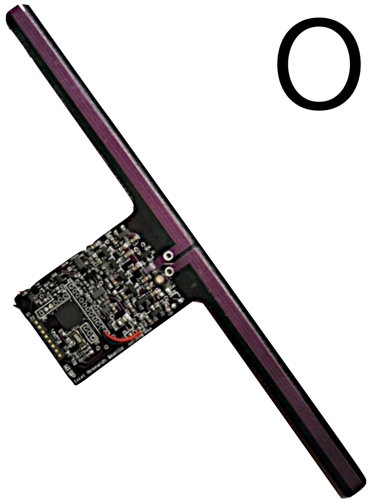
HotPower '08

Getting Things Done ☑ on Computational RFIDs with Energy-Aware Checkpointing and Voltage-Aware Scheduling



Benjamin Ransford, Shane Clark,
Mastooreh Salajegheh, Kevin Fu

Department of Computer Science
University of Massachusetts Amherst

Scenario: RFID Sensor Network

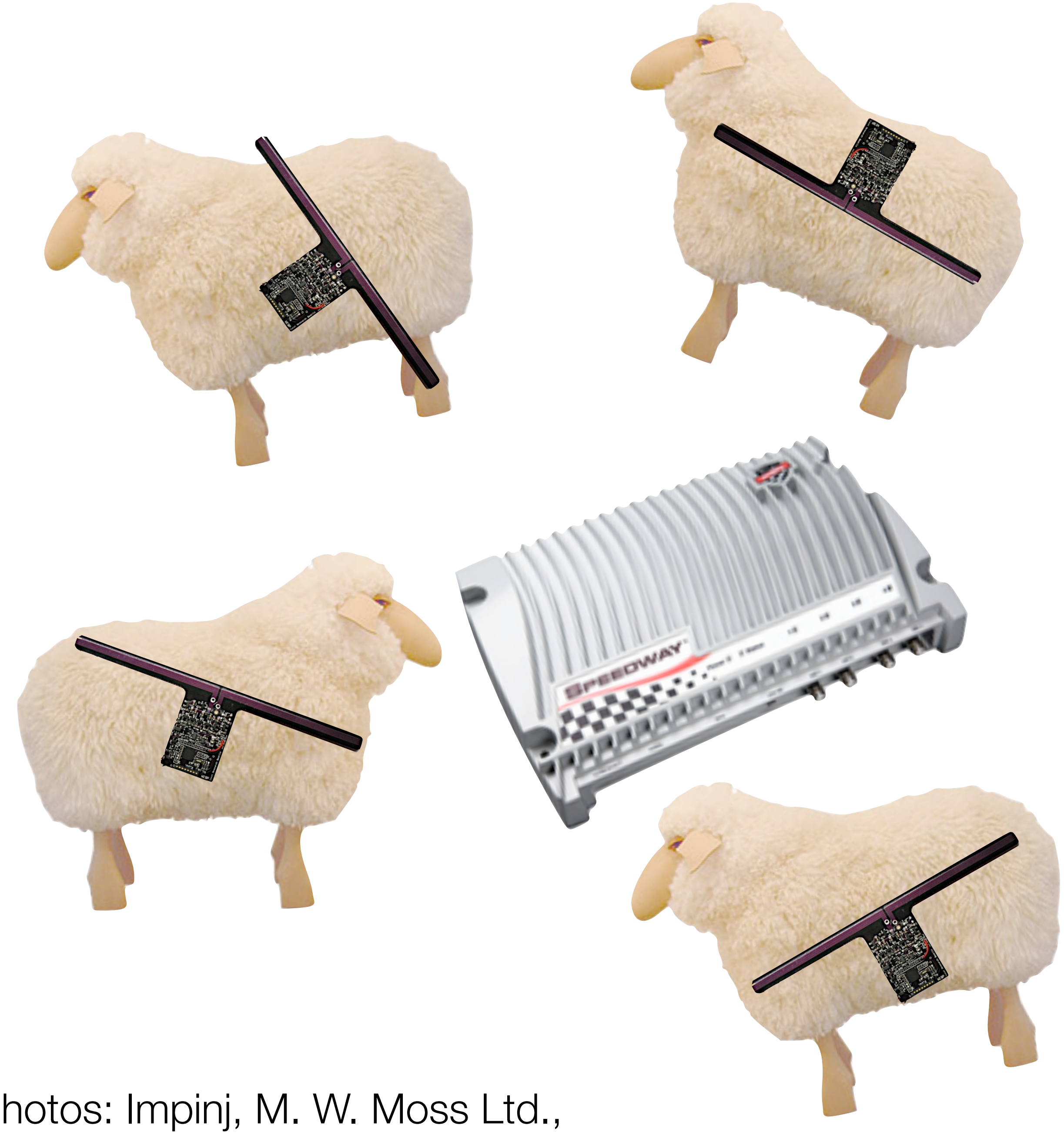
[HotNets '08]

- Maintenance-free
- Batteryless nodes
- RF power harvesting
- Try to do public-key crypto.

Photos: Impinj, M. W. Moss Ltd., reinforcedearth.com

Scenario:
RFID Sensor Network

[HotNets ’08]



- Maintenance-free
- Batteryless nodes
- RF power harvesting
- Try to do public-key crypto.

Photos: Impinj, M. W. Moss Ltd., reinforcedearth.com

Scenario:
RFID Sensor Network

[HotNets '08]

- Maintenance-free
- Batteryless nodes
- RF power harvesting
- Try to do public-key crypto.

Photos: Impinj, M. W. Moss Ltd., reinforcedearth.com

The next 15 minutes

1. Batteryless computing with computational RFID (CRFID)
2. Obstacles to computing on harvested energy
   - Fluctuating supply, power loss
3. Mementos: s/w for *getting things done*
   - Checkpointing, program reordering

Batteries constrain design.

Big & heavy relative to circuits.

Must be replaced or recharged.

Energy density *slooooowly* increasing.
(1991: 204 Wh/l ... 2005: 514 Wh/l)

photos: Duracell, Micro-Tools.com,
Crossbow

How can we do useful computation without a battery?

How can we do useful computation without a battery?

Focus on energy harvesting.

Perils of RF harvesting

- Devices become dependent on energy supply
- Unpredictable supply
- Fluctuating voltage
- Frequent loss of power/state

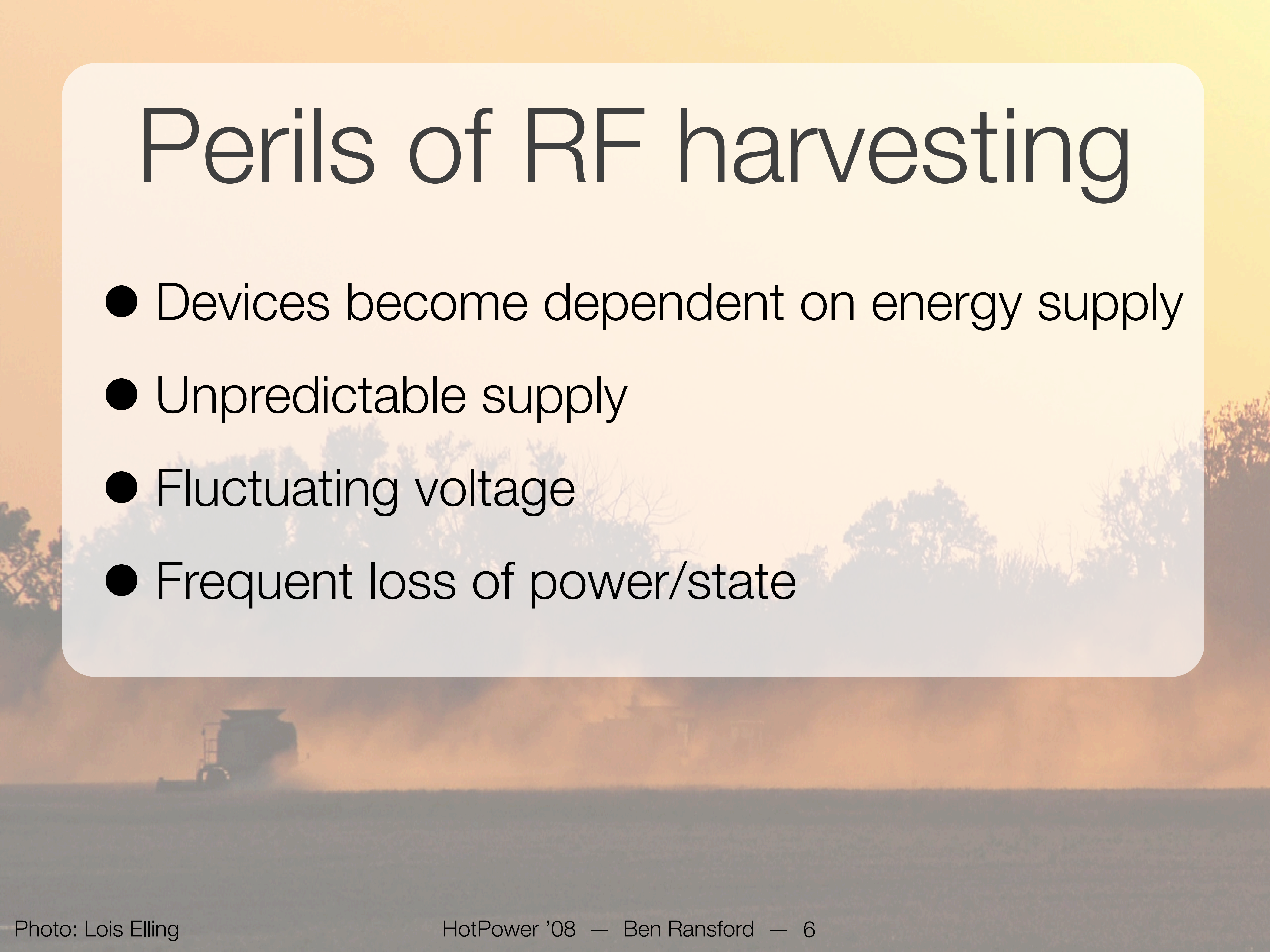


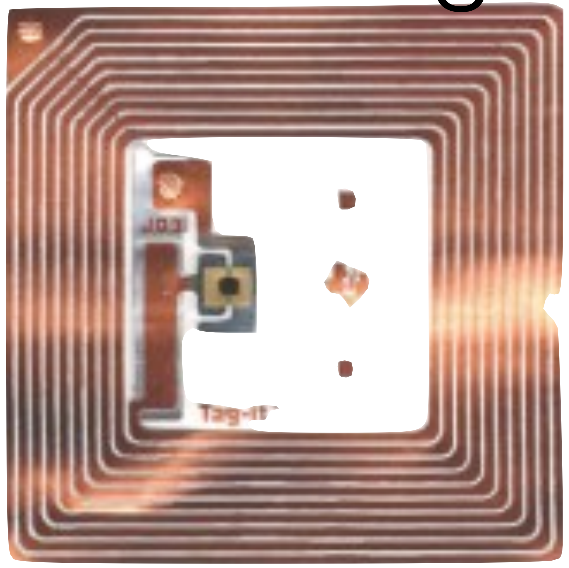
Photo: Lois Elling

Today's batteryless computers

smart card

must finish in one energy lifecycle

RFID tag



non-programmable circuitry

Photos: thisismoney.co.uk, TI

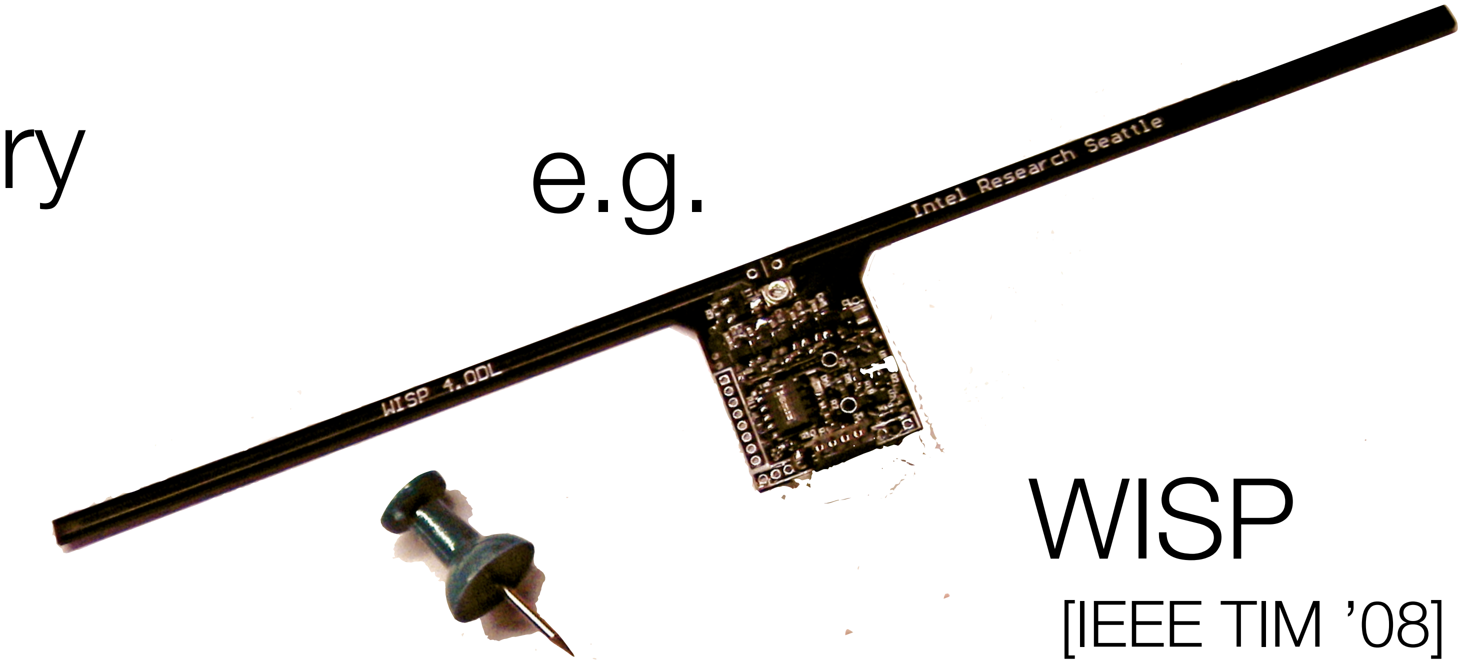
Computational RFID

(new term)

- Modern ultra-low-power (1.5μA sleep, 600μA active) programmable microcontroller
- von Neumann architecture
- RAM, flash memory

No battery...
RF harvesting.

e.g.



WISP
[IEEE TIM '08]

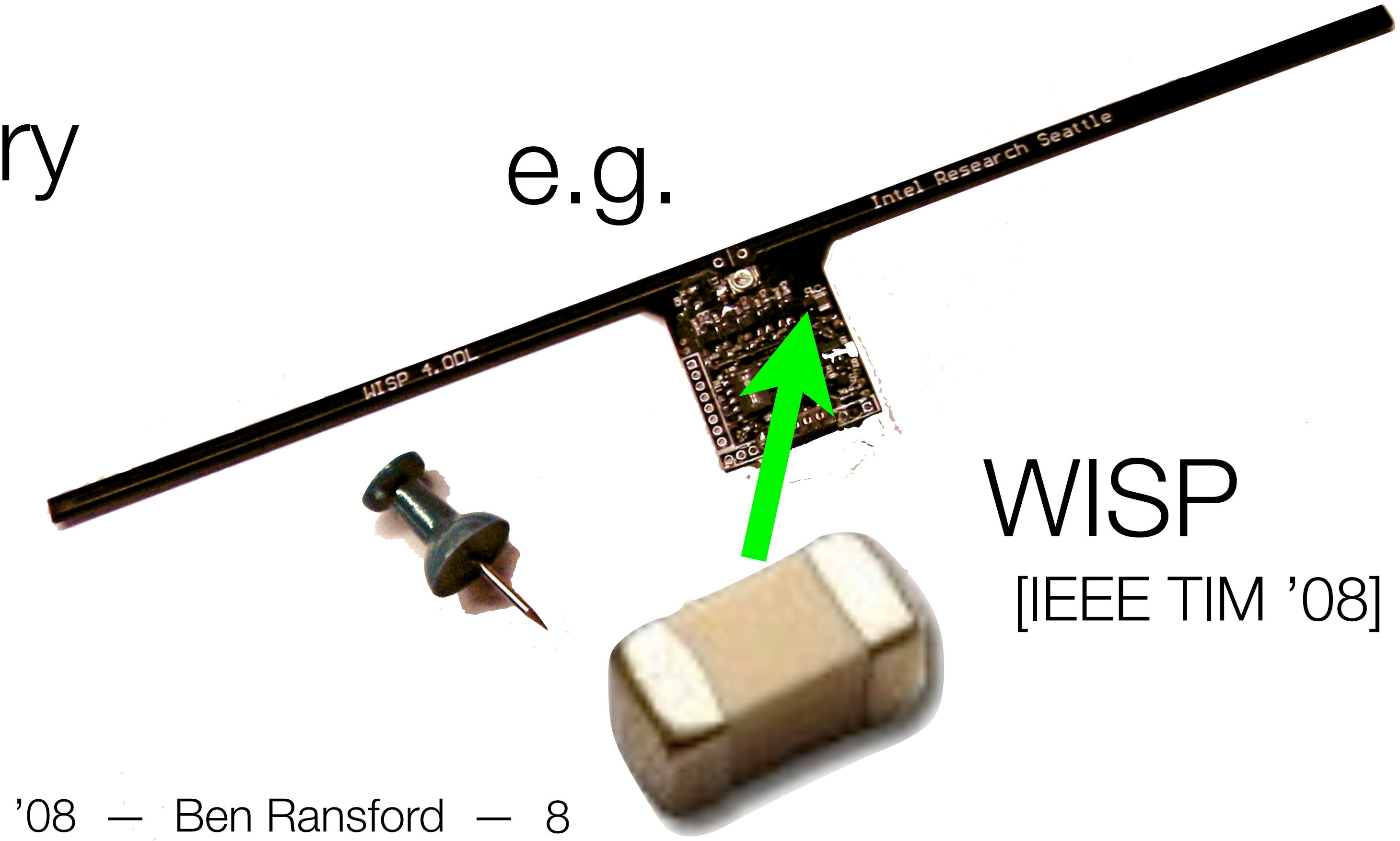
Computational RFID

(new term)

- Modern ultra-low-power (1.5µA sleep, 600µA active) programmable microcontroller
- von Neumann architecture
- RAM, flash memory

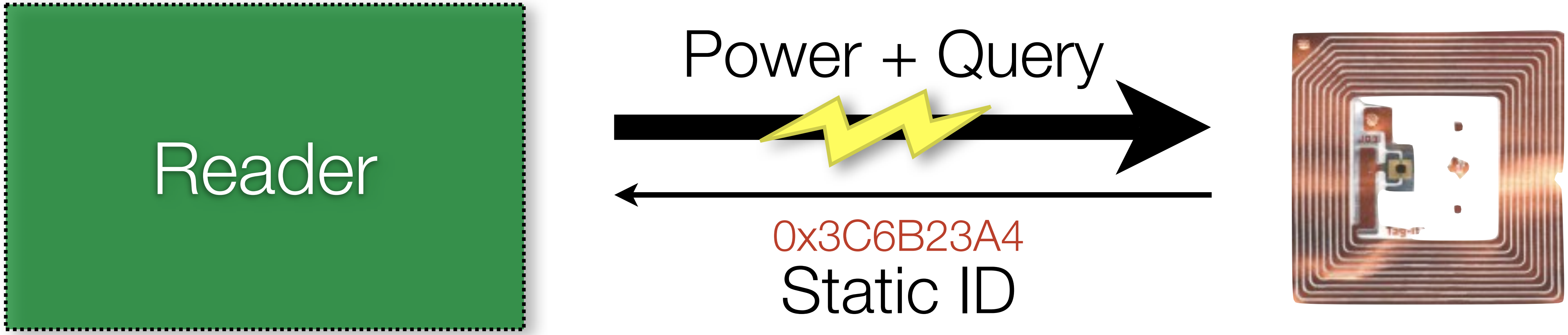
No battery...
RF harvesting.

e.g.



WISP
[IEEE TIM '08]

RFID:



Reader

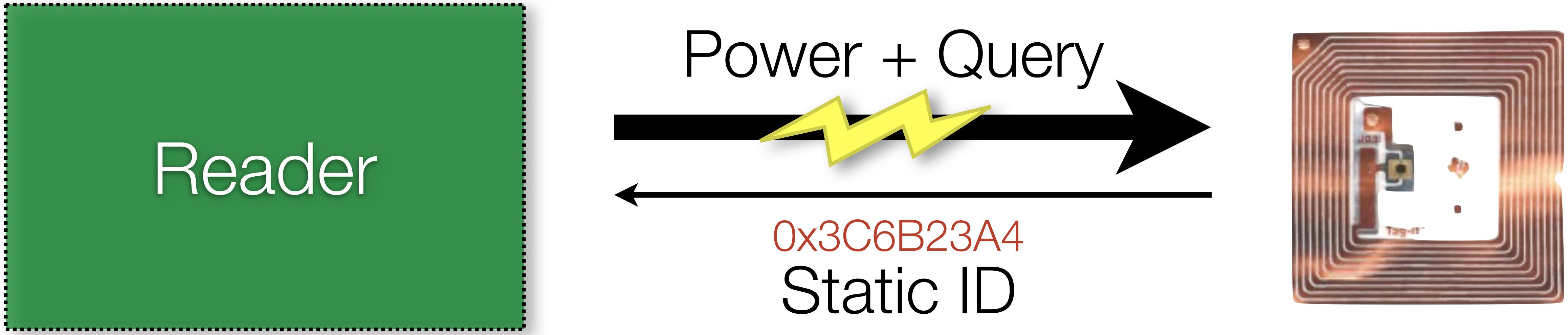
Power + Query

0x3C6B23A4

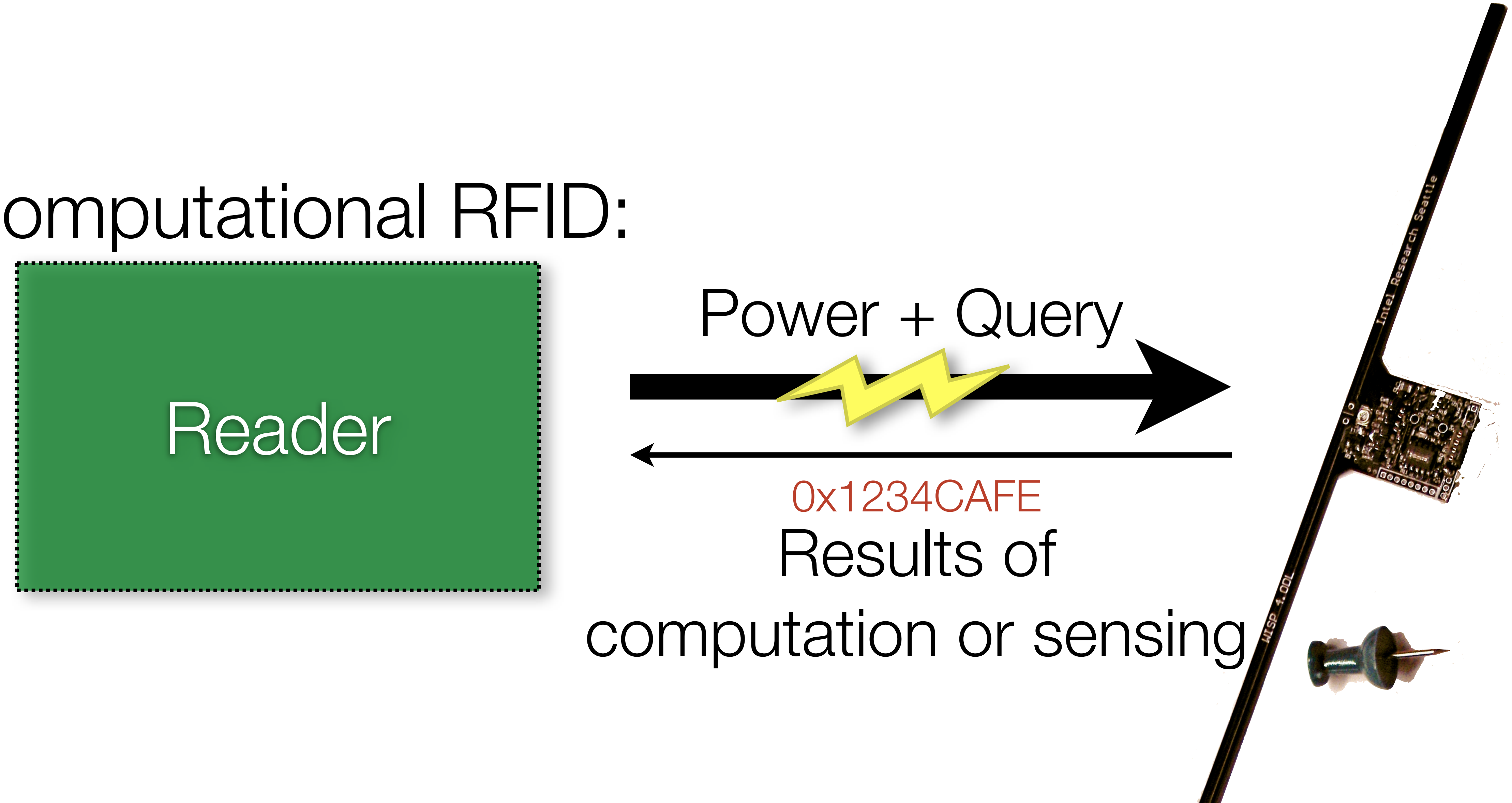
Static ID

Computational RFID:

RFID:



Computational RFID:



Perils of RF harvesting

- Devices become dependent on energy supply
- Unpredictable supply
- Fluctuating voltage
- Frequent loss of power/state

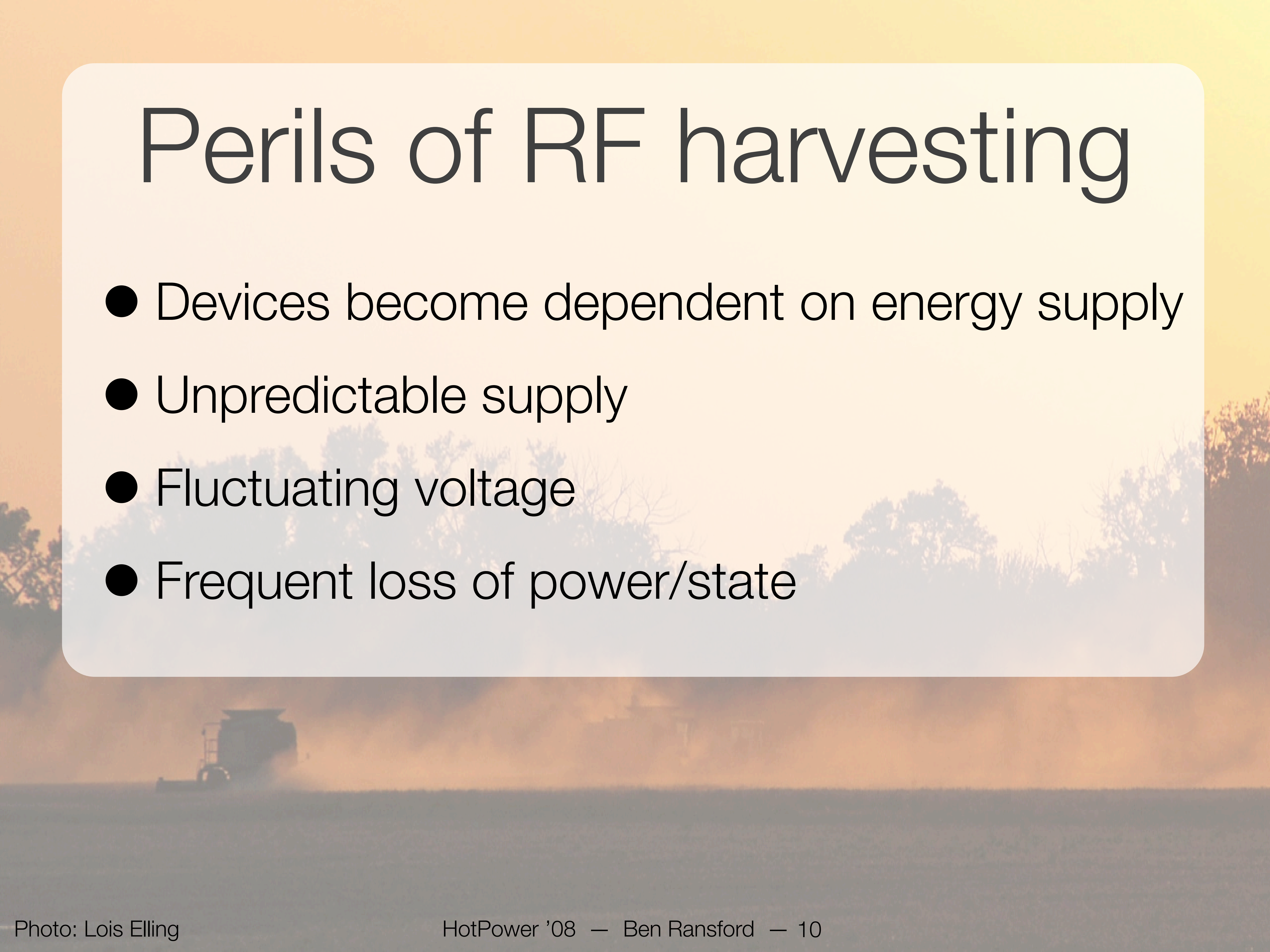


Photo: Lois Elling

Perils of RF harvesting

- Devices become dependent on energy supply
- Unpredictable supply
- Fluctuating voltage
- Frequent loss of power/state

We can address these.

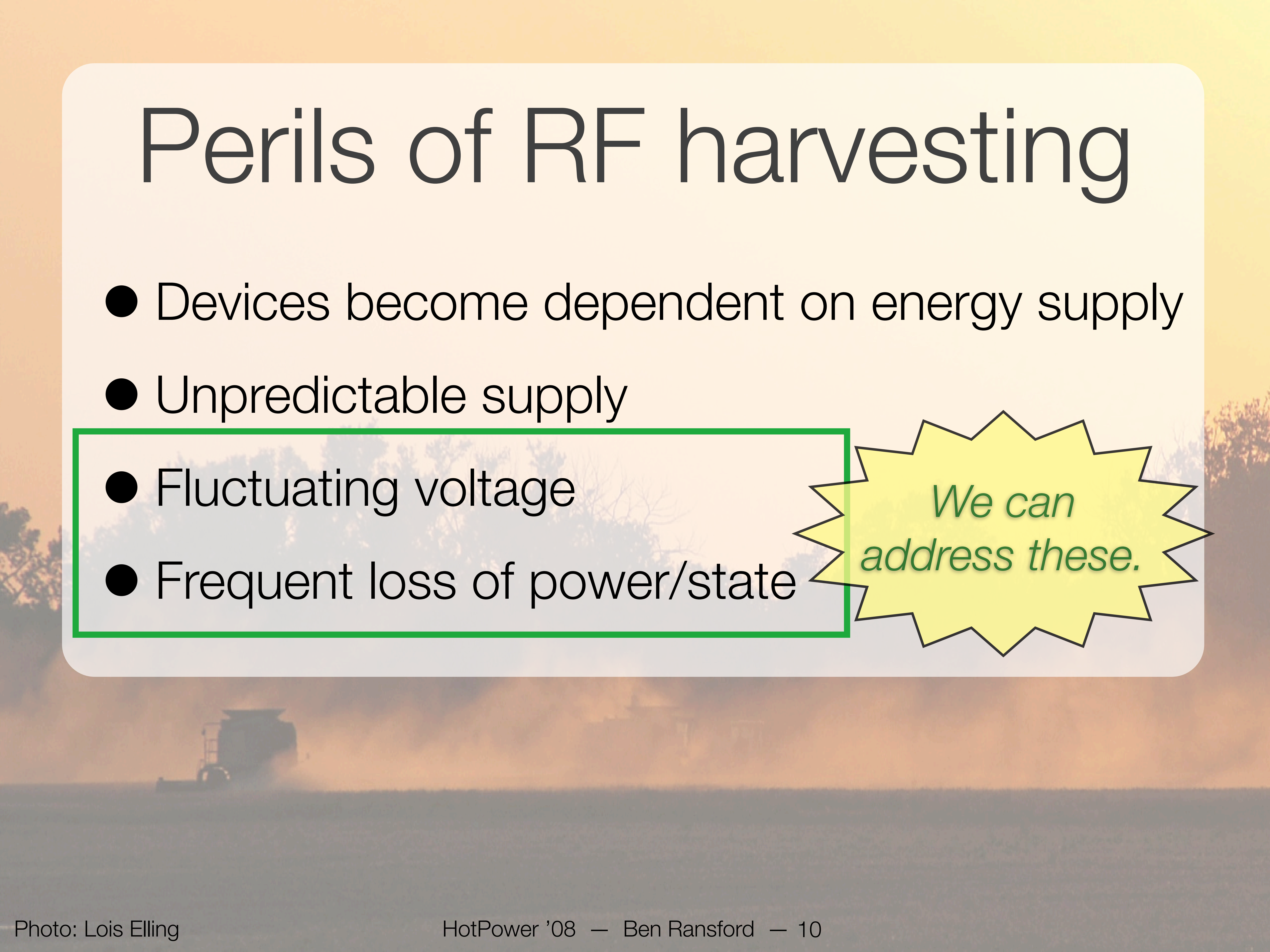


Photo: Lois Elling

Getting things done

Major goal: help programs on CRFIDs make **forward progress** despite fluctuating voltage and constant interruption.

Our system: *Mementos*

- Designed to aid forward progress.

- Execution checkpointing (suspend, resume)
- Program reordering

Our system: *Mementos*

- Designed to aid forward progress.
  - Frequent loss of power/state
- Execution checkpointing (suspend, resume)
- Program reordering

Our system: *Mementos*

- Designed to aid forward progress.

- Frequent loss of power/state

- Execution checkpointing (suspend, resume)
- Program reordering

- Fluctuating voltage

Checkpointing

- Frequent loss of power/state

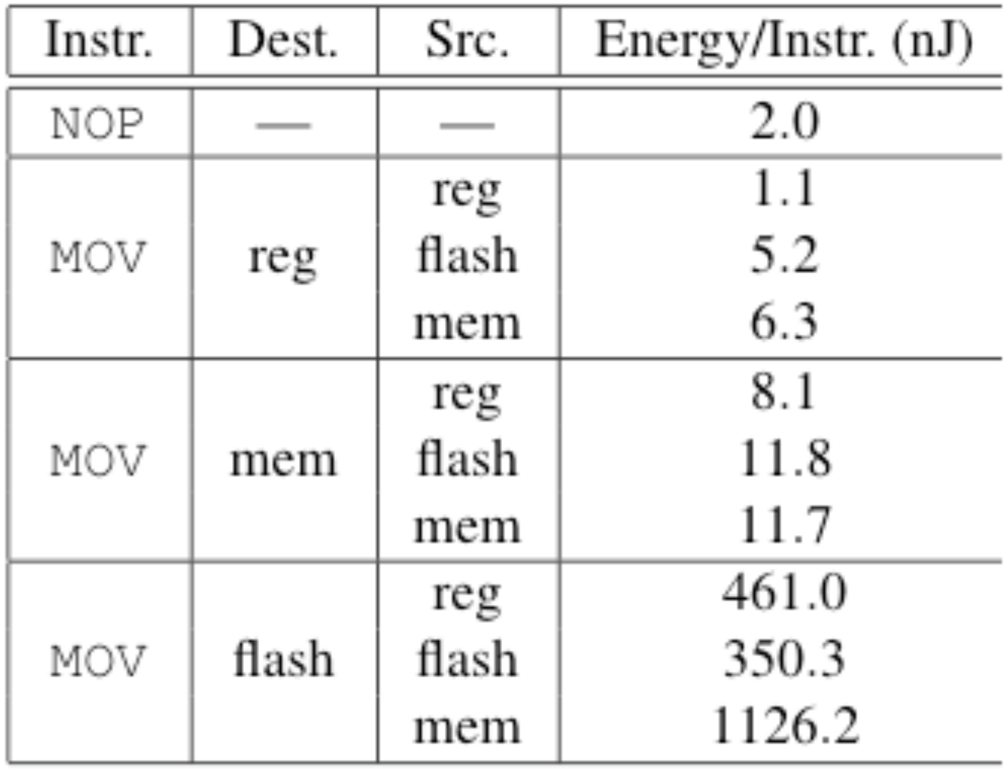
- Idea: save state to flash before dying
- Problem: flash writes consume significant energy when it’s least available.
  - Flash vs. register: 400x more energy
  - Flash vs. memory: 40x more energy

Checkpointing

- Compile time static analysis:
  - Compute per-block energy estimates
- Run time:
  - CRFID checks own voltage
  - Dynamic checkpointing decision

Energy estimation

at compile time



Instr.	Dest.	Src.	Energy/Instr. (nJ)
NOP	—	—	2.0
MOV	reg	reg	1.1
		flash	5.2
		mem	6.3
MOV	mem	reg	8.1
		flash	11.8
		mem	11.7
MOV	flash	reg	461.0
		flash	350.3
		mem	1126.2

Platform-specific energy profile

```
label1:
MOV R11, R12        1 nJ
ADD R12, R8         1 nJ
(Flash write)       461 nJ
JMP label2          --
...                 ...
```

Annotated instruction stream

e.g.: modexp

- Halve 32-bit exponent, square 32-bit base
    - No checkpointing: dies before finishing

e.g.: modexp

- Halve 32-bit exponent, square 32-bit base
  - No checkpointing: dies before finishing

e.g.: modexp

- Halve 32-bit exponent, square 32-bit base
  - No checkpointing: dies before finishing
  - Checkpoint halfway through:
    - Save base, exp., accumulated result after 15 iterations; die before finishing
    - Restore from checkpoint; 17 more iterations; complete.

Program reordering

- Fluctuating voltage

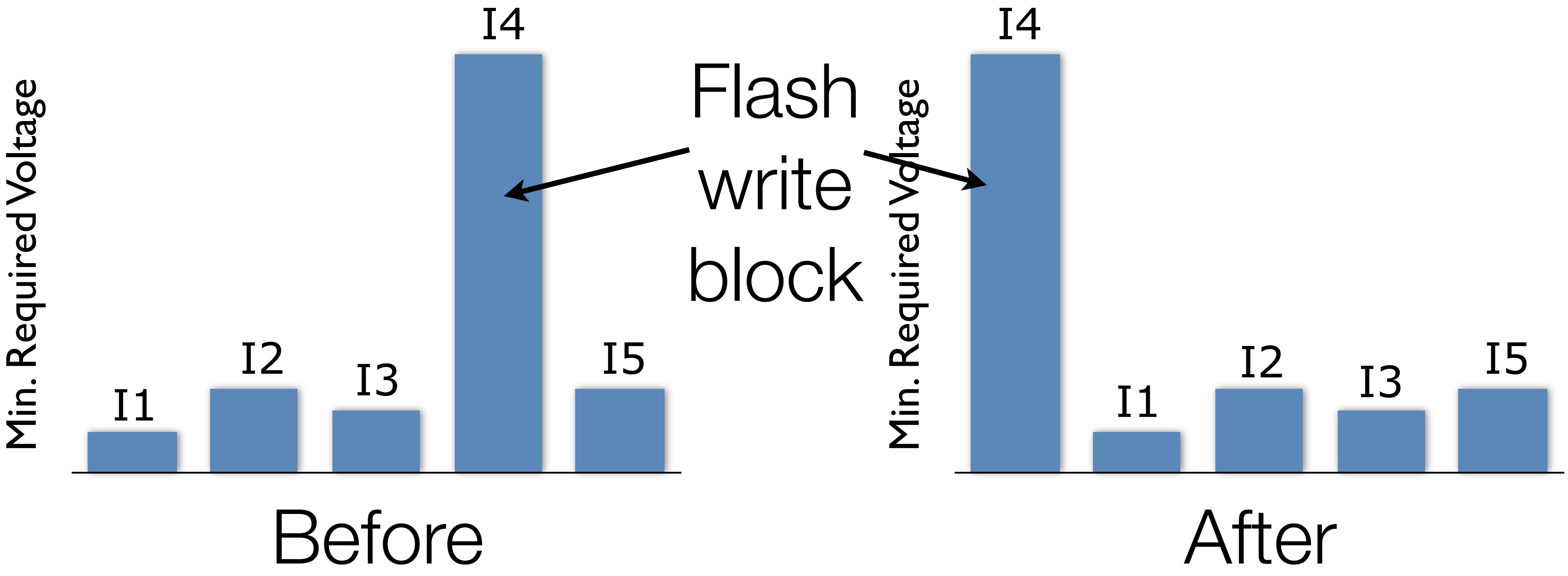
- Observations:
  - Some operations require higher voltage
  - Voltage tends to decline during each device lifecycle
  - Microcontrollers don’t like continuously varying voltage (PLL logic limitations)

Program reordering

- Static analysis at compile time
  - Estimate energy requirements
  - Derive dependency graph
- Must not violate program semantics!

Program reordering

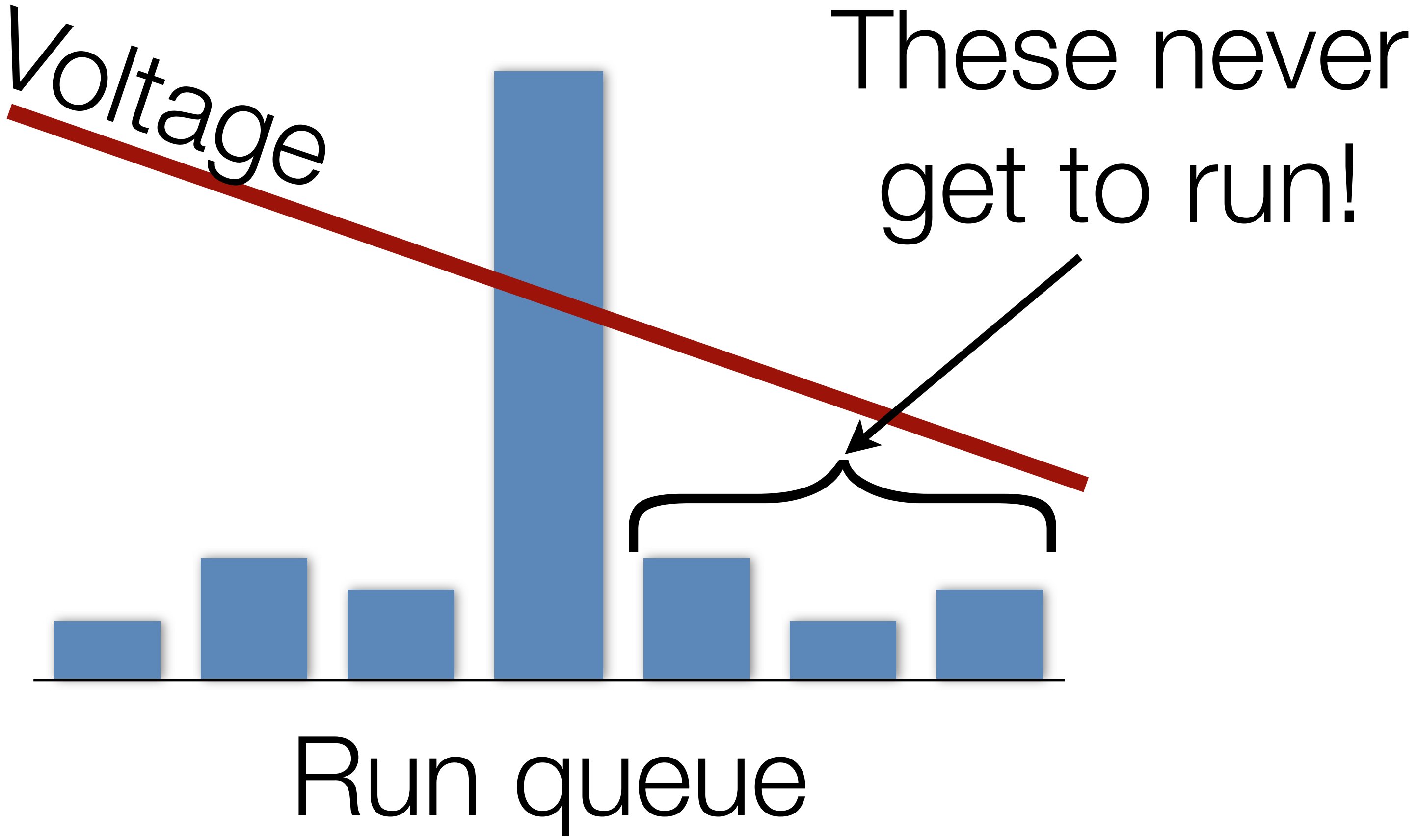
- Voltage declines: reorder independent code chunks at **compile time** to execute high-V ops when voltage is high



Flash write block

Program reordering

- Smaller timescale: adaptively reschedule program chunks at **run time** to avoid logjams



Challenges

- Predicting program behavior is hard.
- Balance checkpointing behavior:
  - How much state to save
  - How often to checkpoint
- Program reordering:
  - Finding dependencies can be hard

Physical barriers

- Can’t harvest RF energy at arbitrary distances (current prototypes: ≤ 10 m)
- Diode drop limits energy harvesting

CRFID applications

- Medical implants [Oakland ’08]
- RFID Sensor Networks [HotNets ’08]

- Computation in inaccessible locations.

  fragile

  hazardous

Future developments

- Our work:
  - Fully implement checkpointing, reordering
  - Device profiling
- CRFIDs:

Intel Looks To Blanket The World With Self-Powered Sensors

By Antone Gonsalves
InformationWeek

December 5, 2008 05:37 PM

- Intel Research competition (Google *intel wisp challenge*)

Summary

- Computational RFIDs: general-purpose batteryless computers
- Mementos for forward progress
  - Checkpointing to cope with constant power interruptions
  - Program reordering to cope with fluctuating voltage

Applications?
Challenges?
Alternatives?

ransford@cs.umass.edu