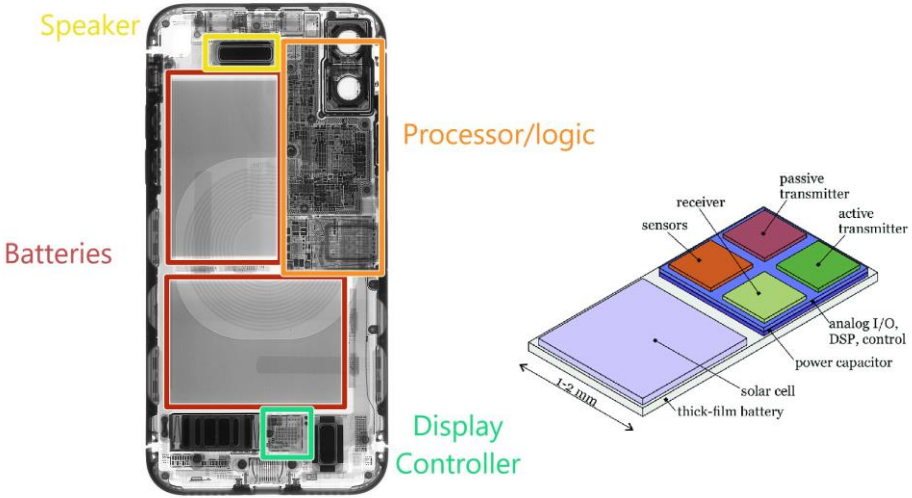
Energy-adaptive Buffering for Efficient, Responsive, and Persistent Batteryless Systems

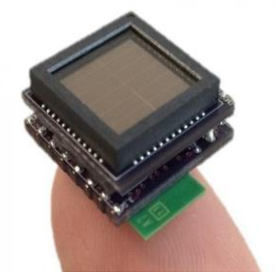
Harrison Williams
hrwill@vt.edu

Matthew Hicks
mdhicks2@vt.edu

COMPUTER SCIENCE
VIRGINIA TECH

Energy harvesting sensor systems





Bulky

Limited

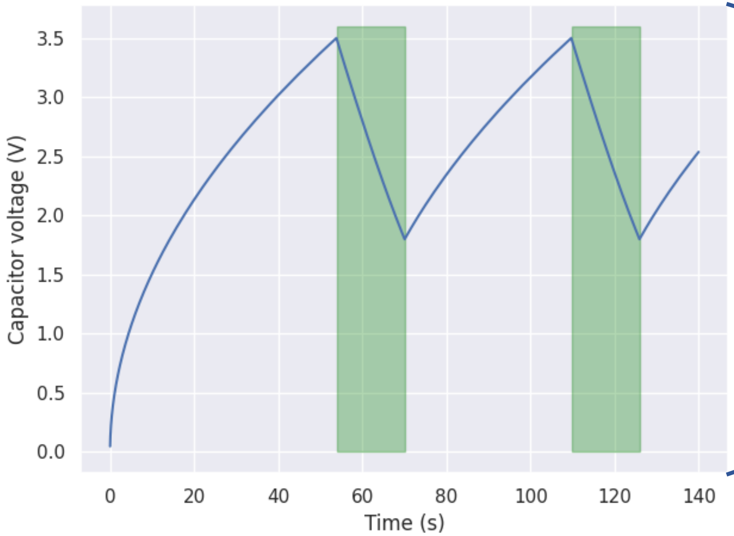
Explosive

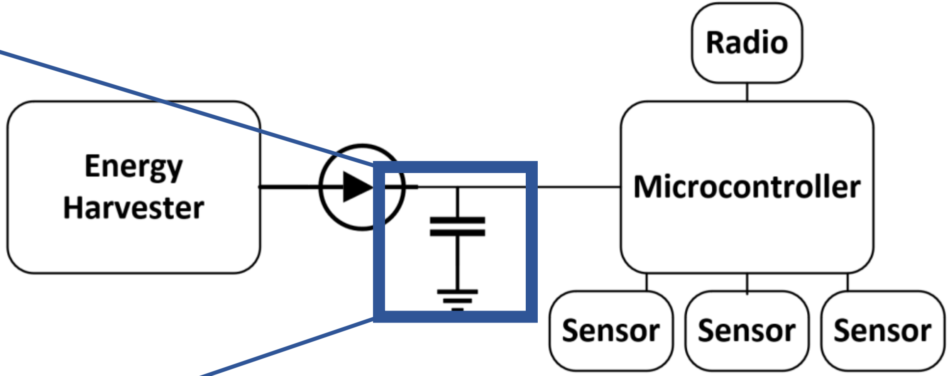
Small

Perpetual

Stable

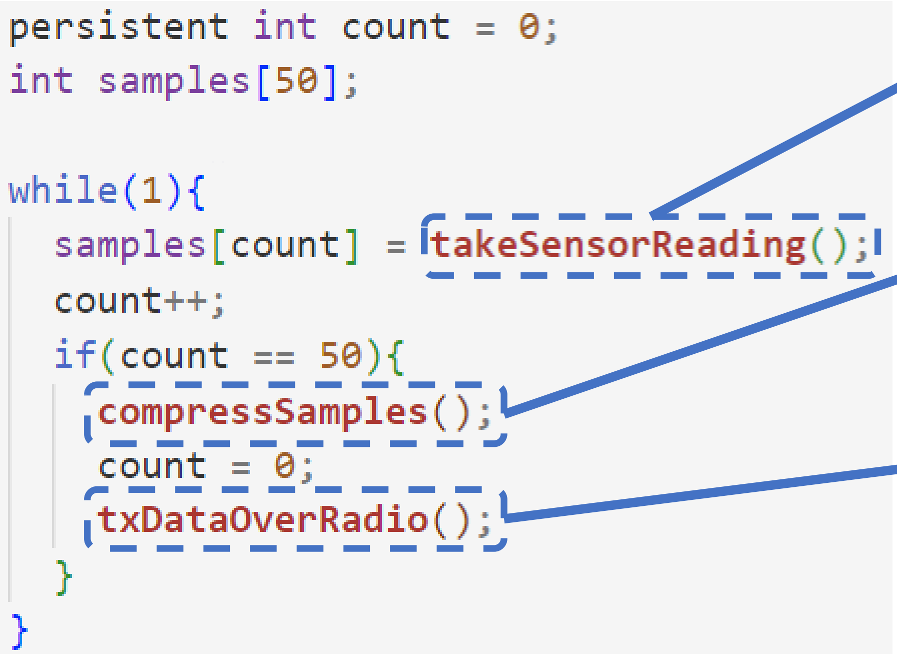
Basic batteryless operation





Capacitor chosen at HW design time

System requirements drive capacitor size



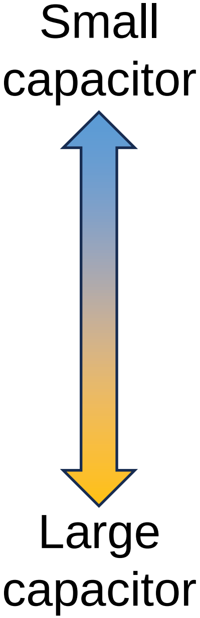
```
persistent int count = 0;
int samples[50];

while(1){
    samples[count] = takeSensorReading();
    count++;
    if(count == 50){
        compressSamples();
        count = 0;
        txDataOverRadio();
    }
}
```

Rapidly charge to enable system

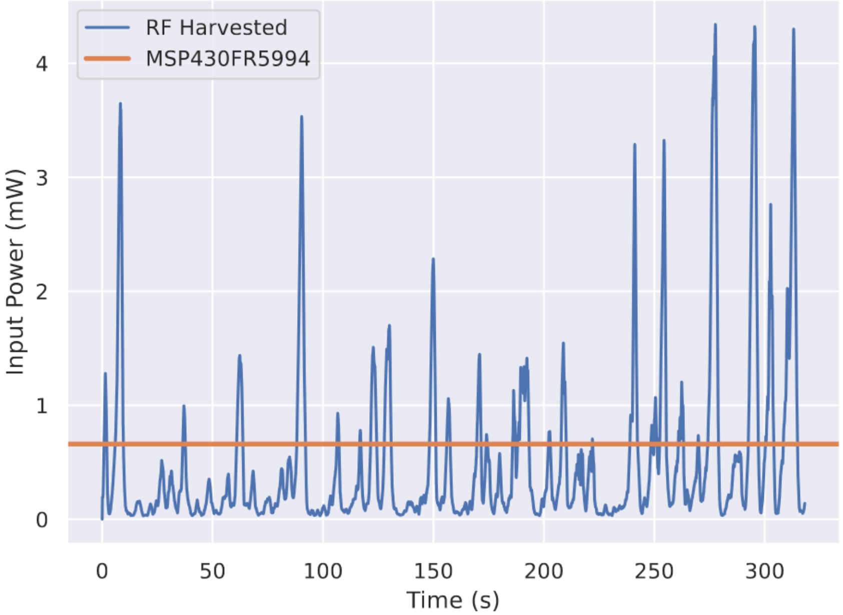
Reduce SW overhead from restarts

Support high-energy atomic operations



Problem: rapid charge time and high capacity are mutually exclusive

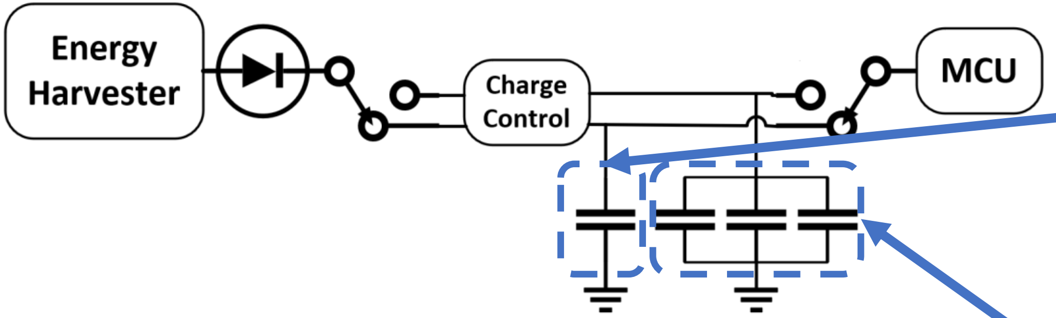
Real world volatility requires capacity

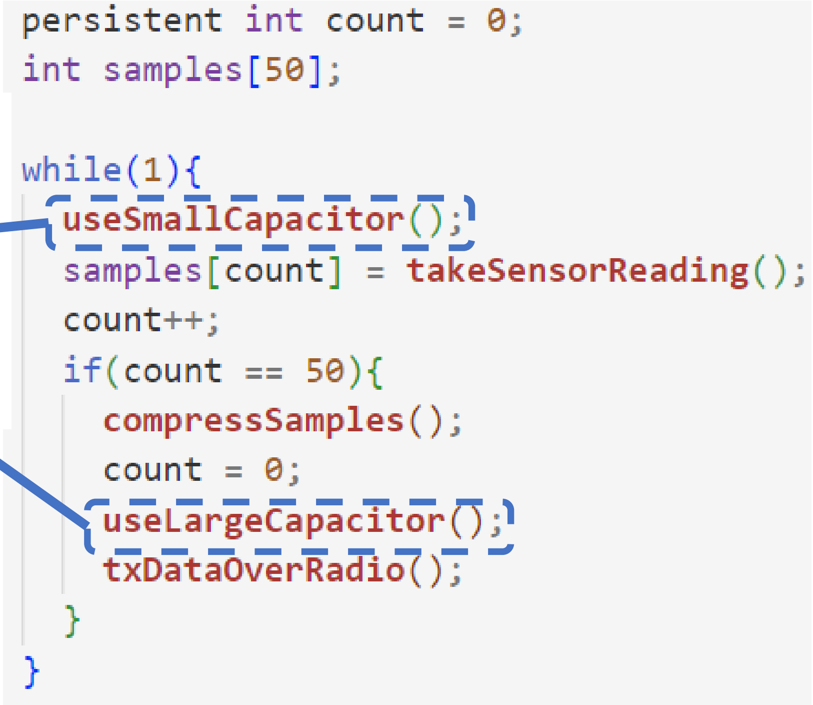


Need capacity to capture power spikes **without** compromising charge time

State of the art: capacitor multiplexing

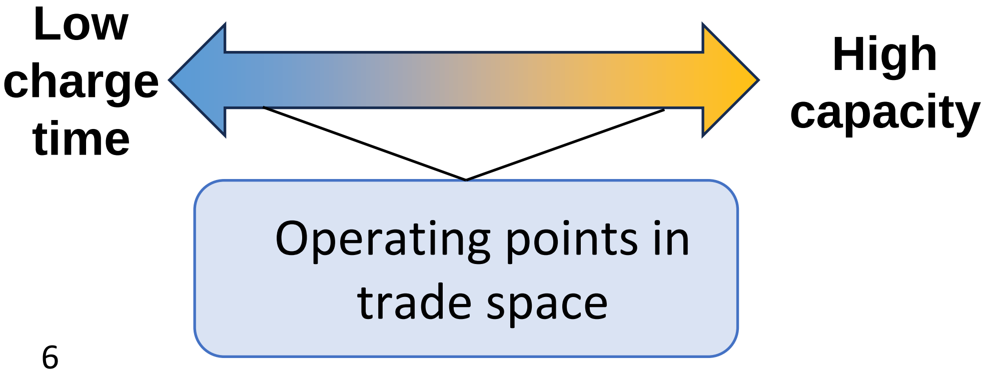
[SenSys ‘15, ASPLOS ‘18]





```
persistent int count = 0;
int samples[50];

while(1){
  useSmallCapacitor();
  samples[count] = takeSensorReading();
  count++;
  if(count == 50){
    compressSamples();
    count = 0;
    useLargeCapacitor();
    txDataOverRadio();
  }
}
```

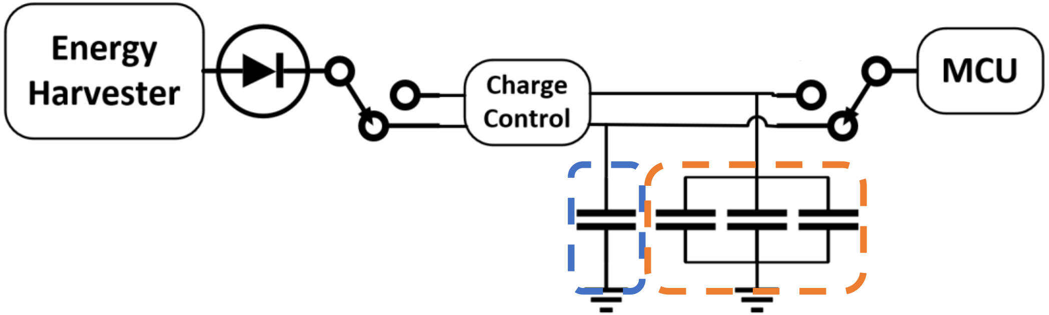


Low charge time

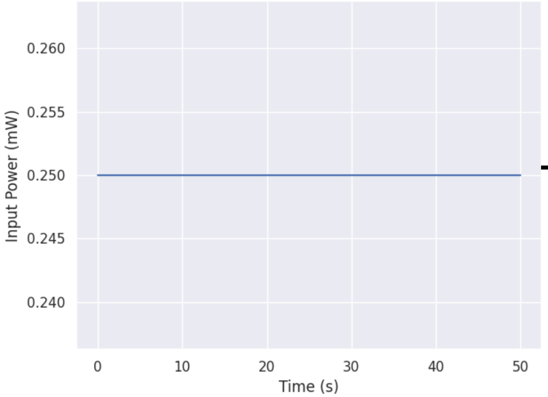
High capacity

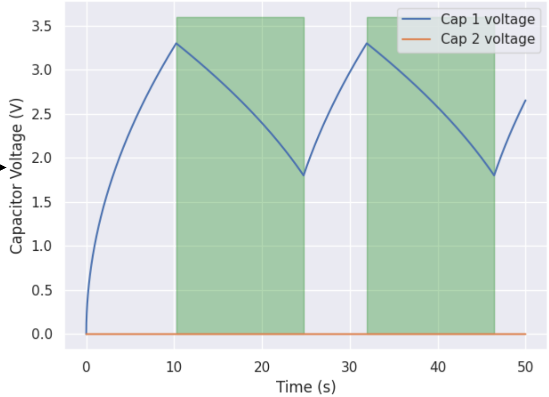
Operating points in trade space

Energy fragmentation reduces performance



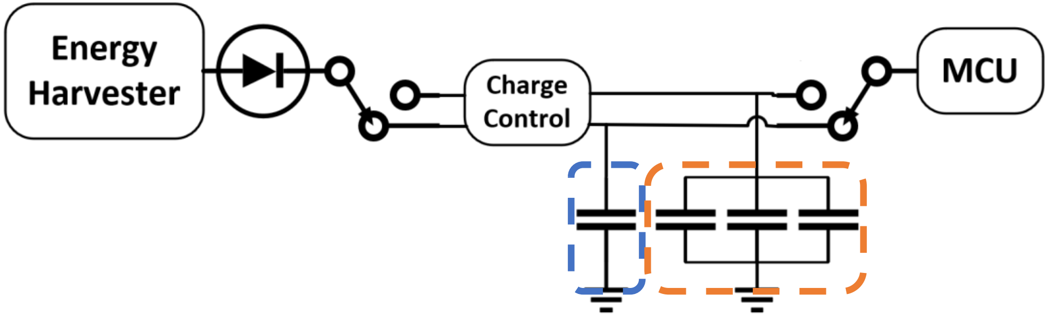
Avg. power: 0.25 mW



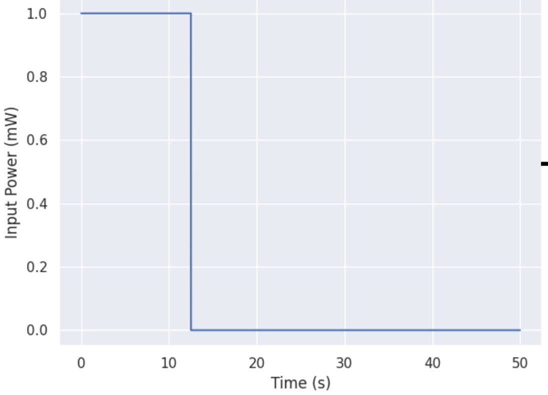


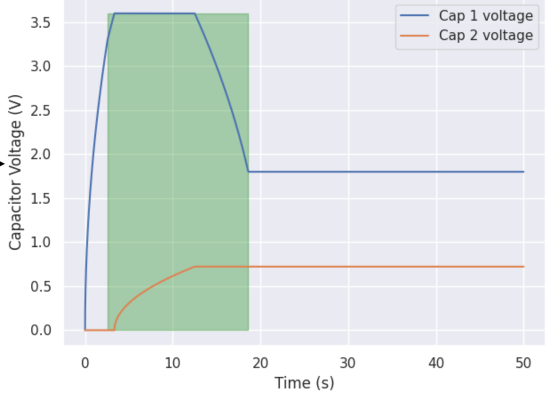
On time: 29 s

Energy fragmentation reduces performance



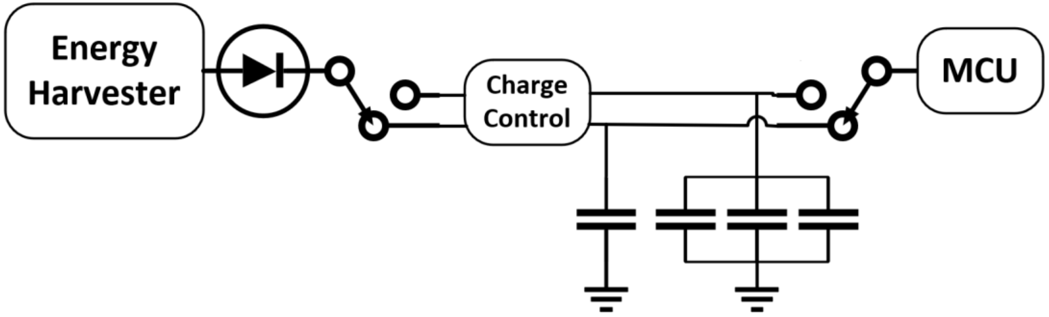
Avg. power: 0.25 mW

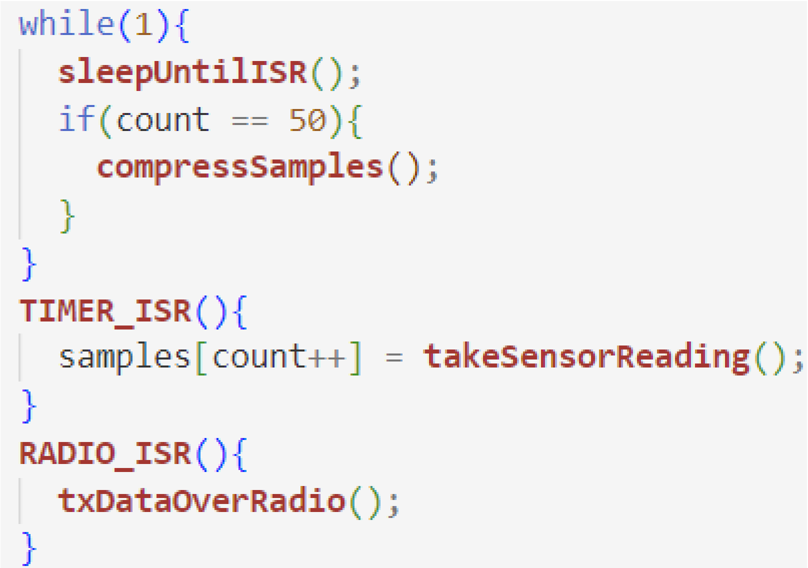




On time: 16 s

Energy fragmentation complicates design



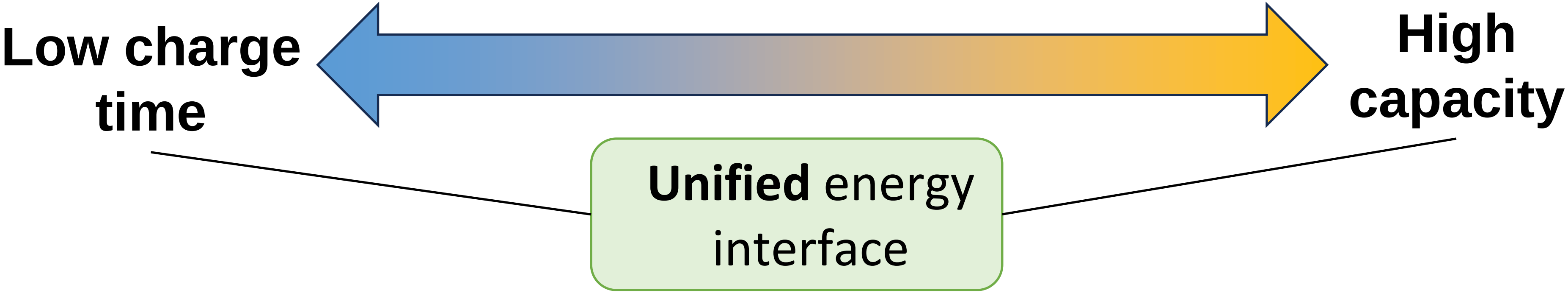

```
while(1){
  sleepUntilISR();
  if(count == 50){
    compressSamples();
  }
}
TIMER_ISR(){
  samples[count++] = takeSensorReading();
}
RADIO_ISR(){
  txDataOverRadio();
}
```

Which capacitor do I charge first?

What if my priorities change?

Energy is not **fungible**

Problem statement



Low charge time

High capacity

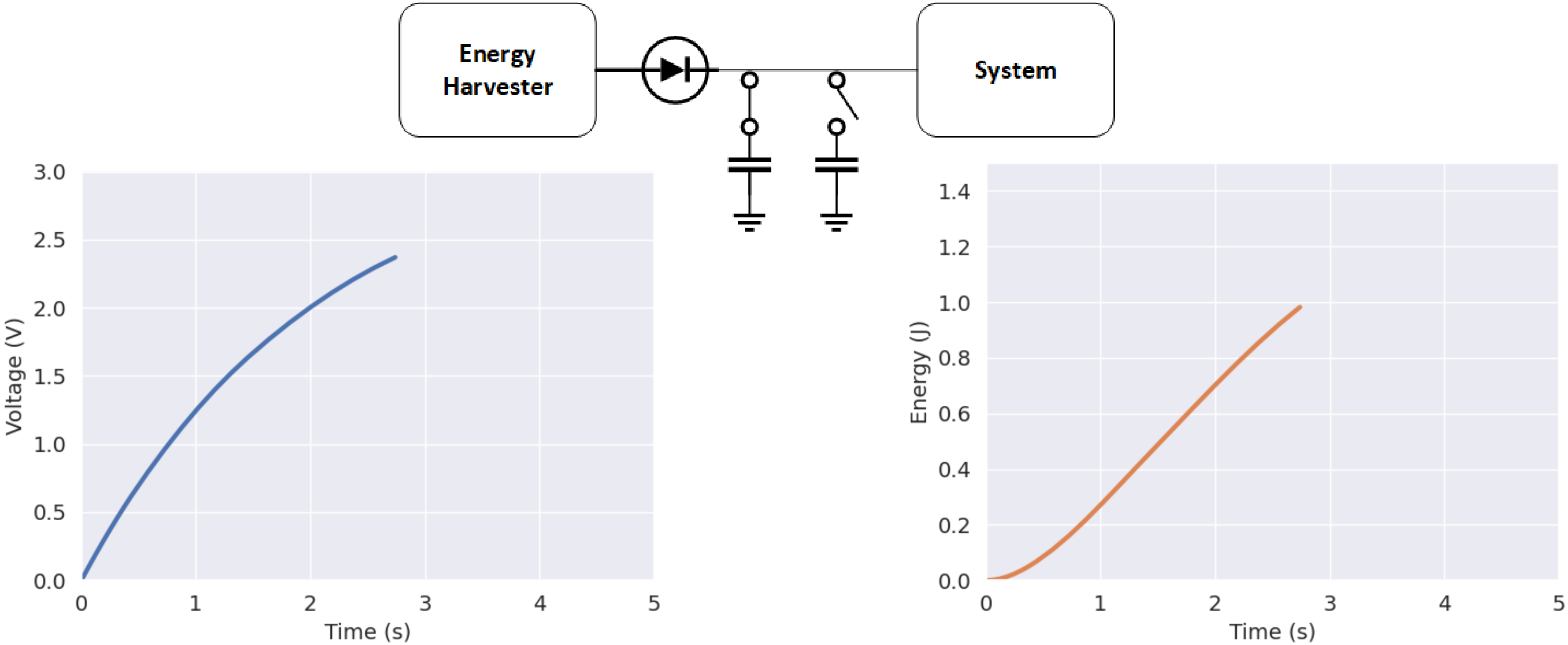
Unified energy interface

✓ Provide reactivity/capacity while maintaining energy **fungibility**

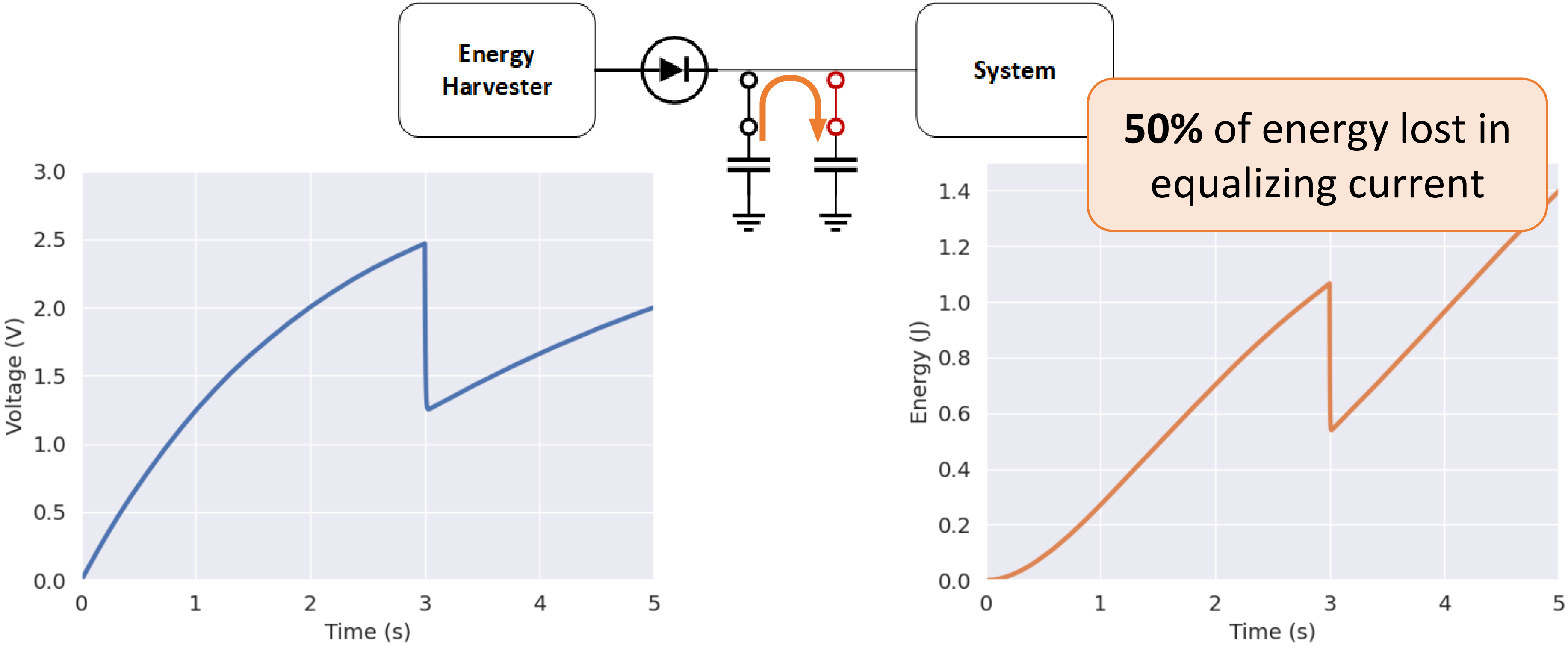
✓ Do the above **without programmer burden**

We need a system that expands and contracts capacitance according to stored energy

Efficient variable capacitance

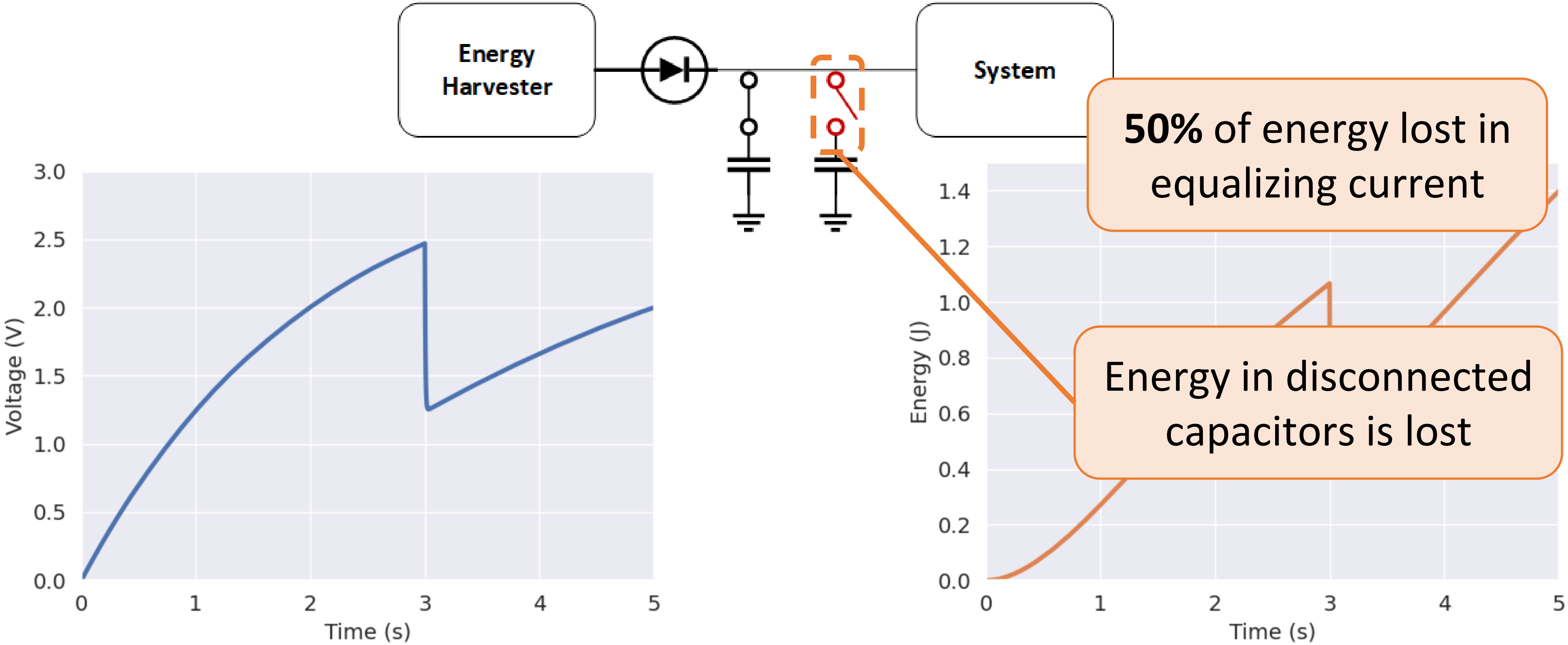


Efficient variable capacitance



50% of energy lost in equalizing current

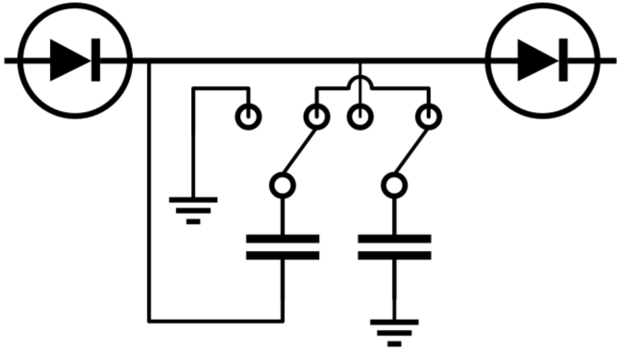
Efficient variable capacitance

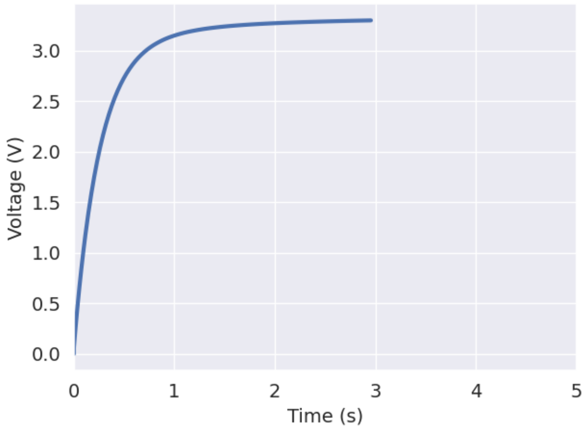


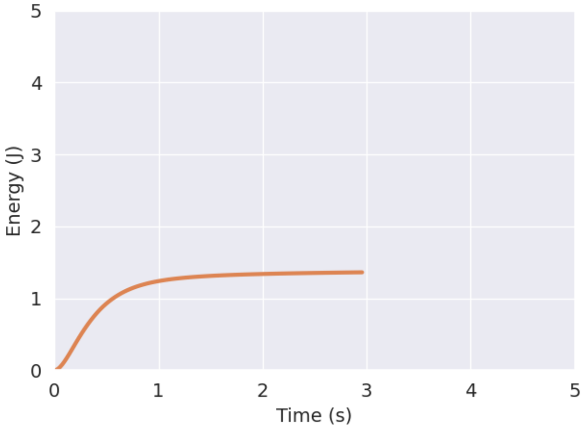
50% of energy lost in equalizing current

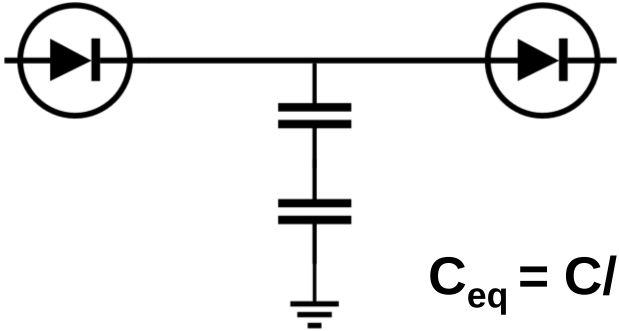
Energy in disconnected capacitors is lost

Efficient variable capacitance



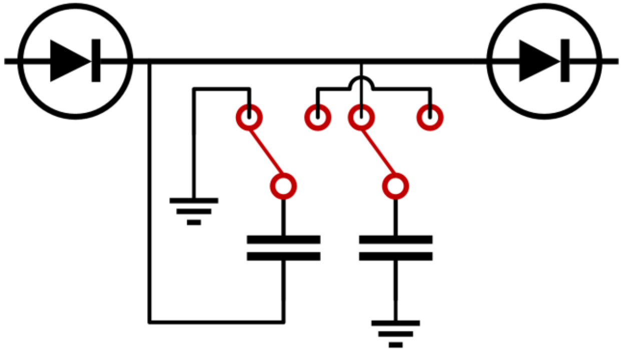


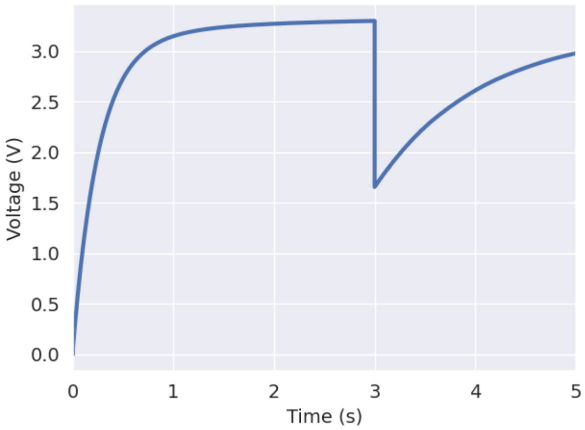


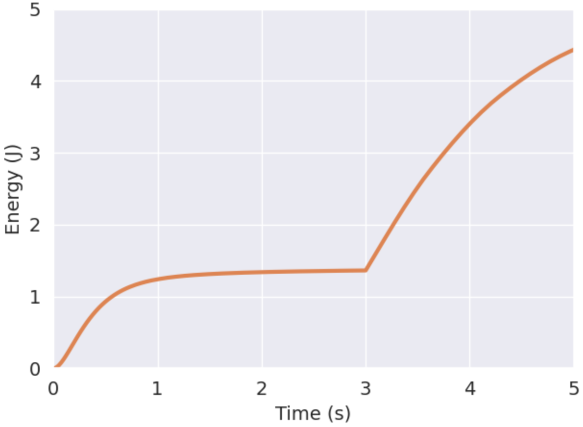


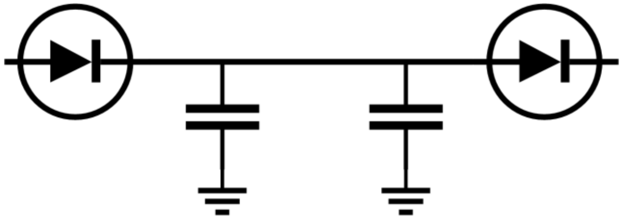
$C_{eq} = C/2$

Efficient variable capacitance





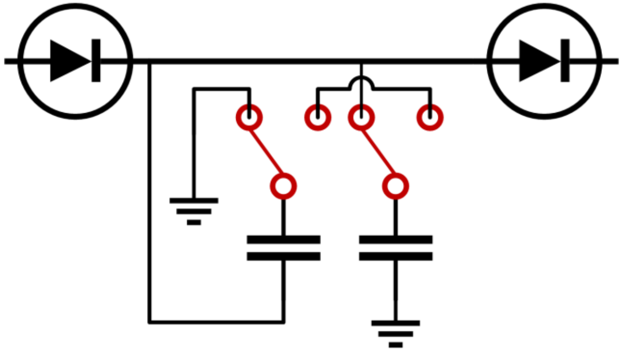


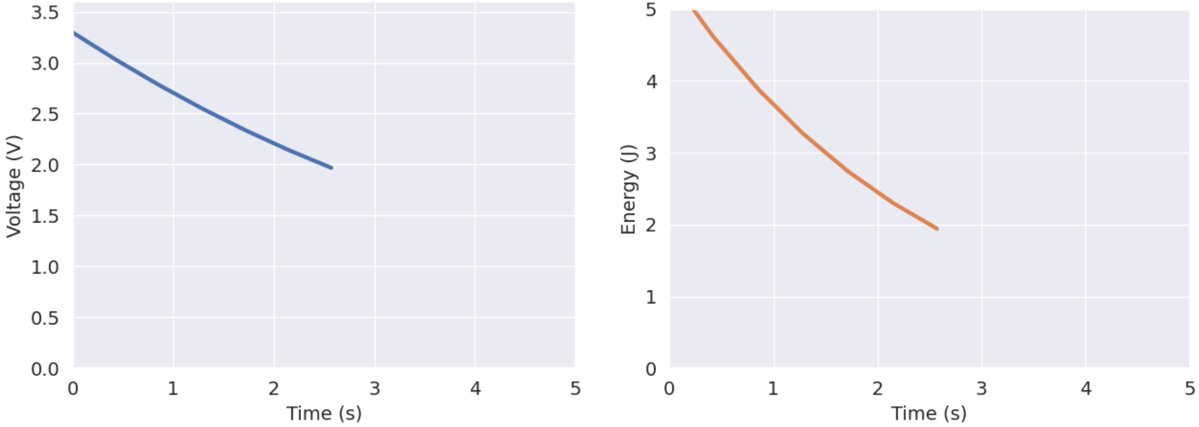


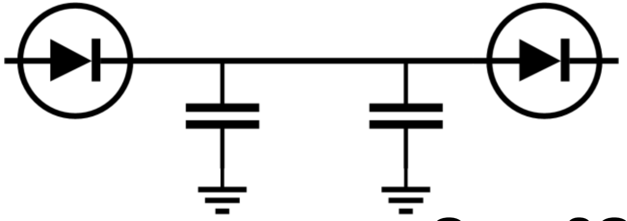
$C_{eq} = 2C$

Variable capacitance/voltage without energy loss

Efficient variable capacitance



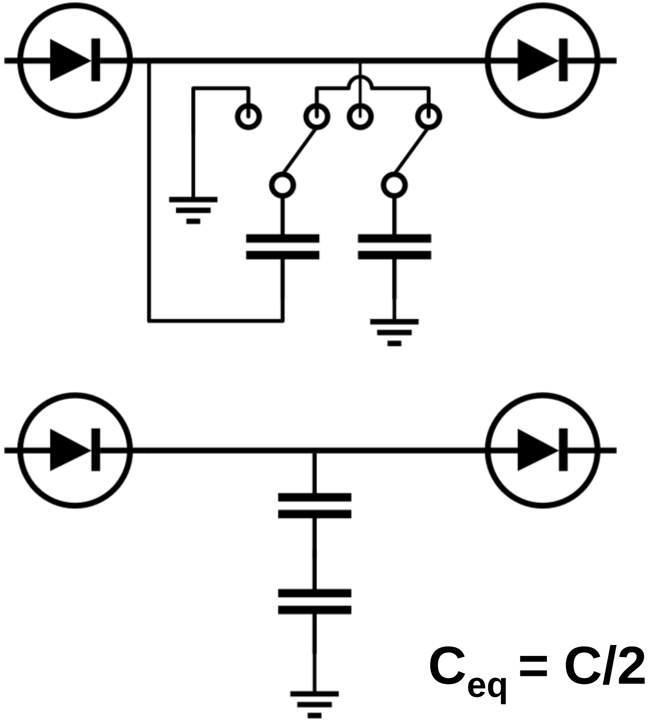


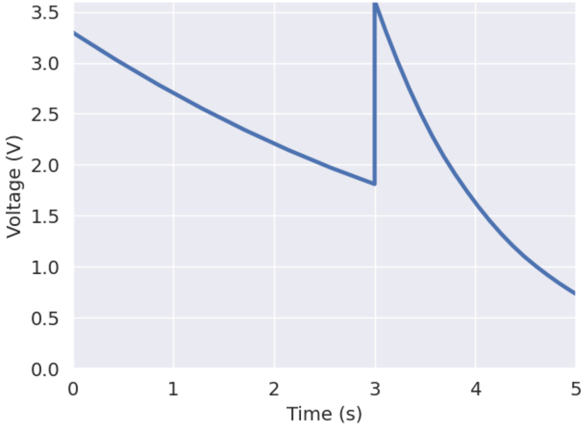


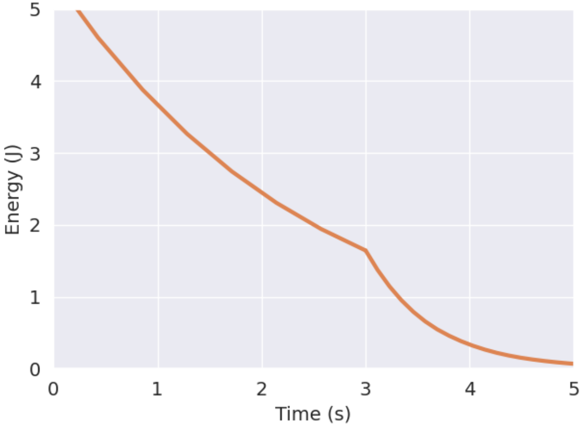
C_{eq} = 2C

Reduce capacitance during discharge

Efficient variable capacitance







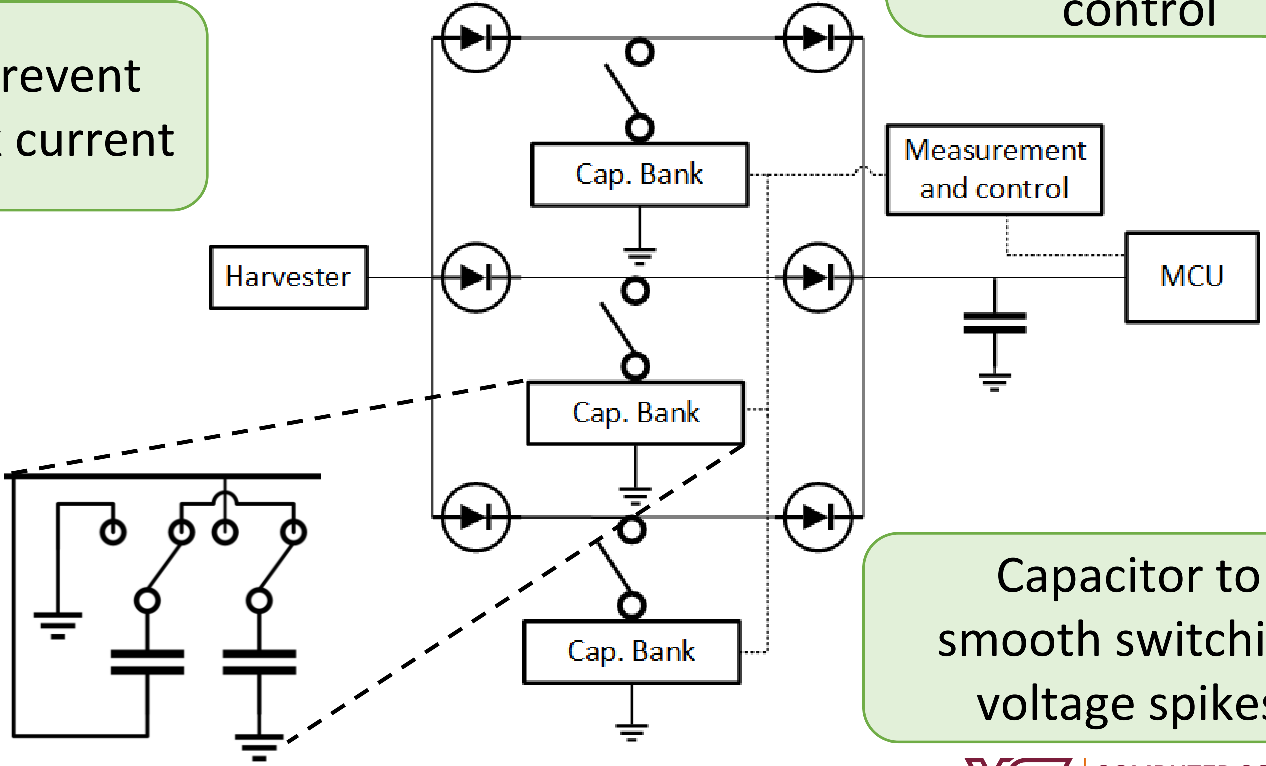
Reduce capacitance during discharge

Increase energy extracted

Our approach: REACT

Diodes prevent inter-bank current

Software interface for capacitance control



Capacitor to smooth switching voltage spikes

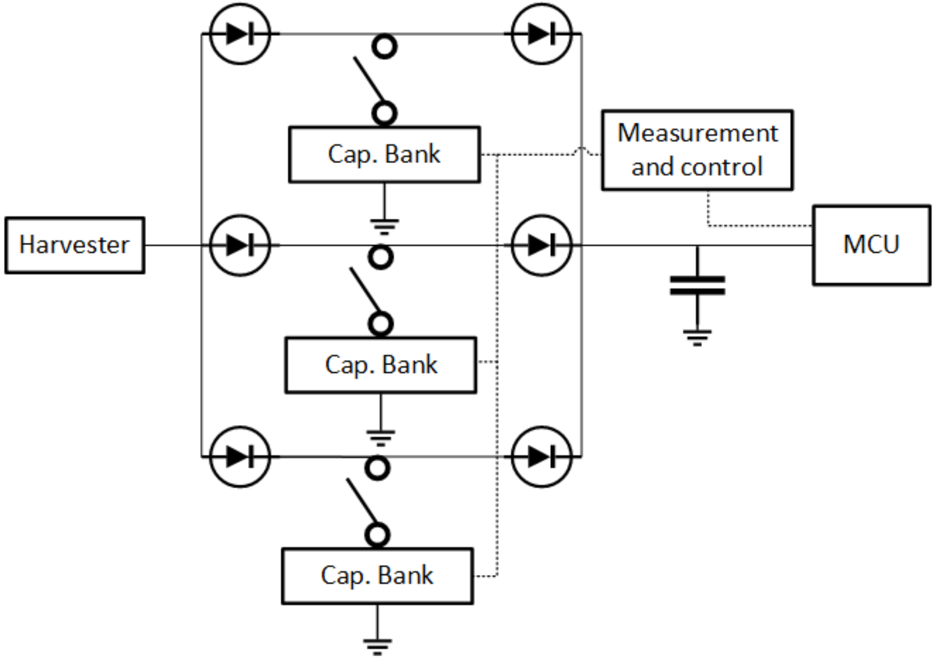
Our approach: REACT

A unified, reconfigurable capacitor architecture

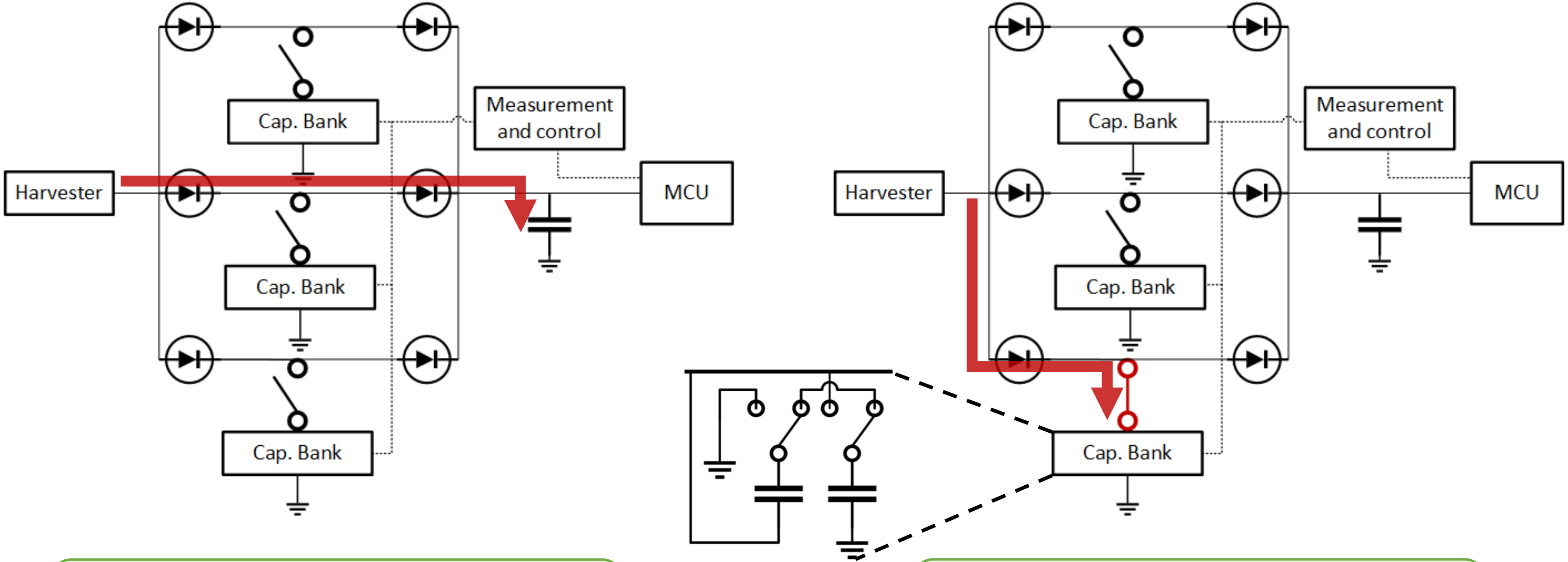
Fast charge **and** high capacity

Fungible energy storage

Transparent software interface



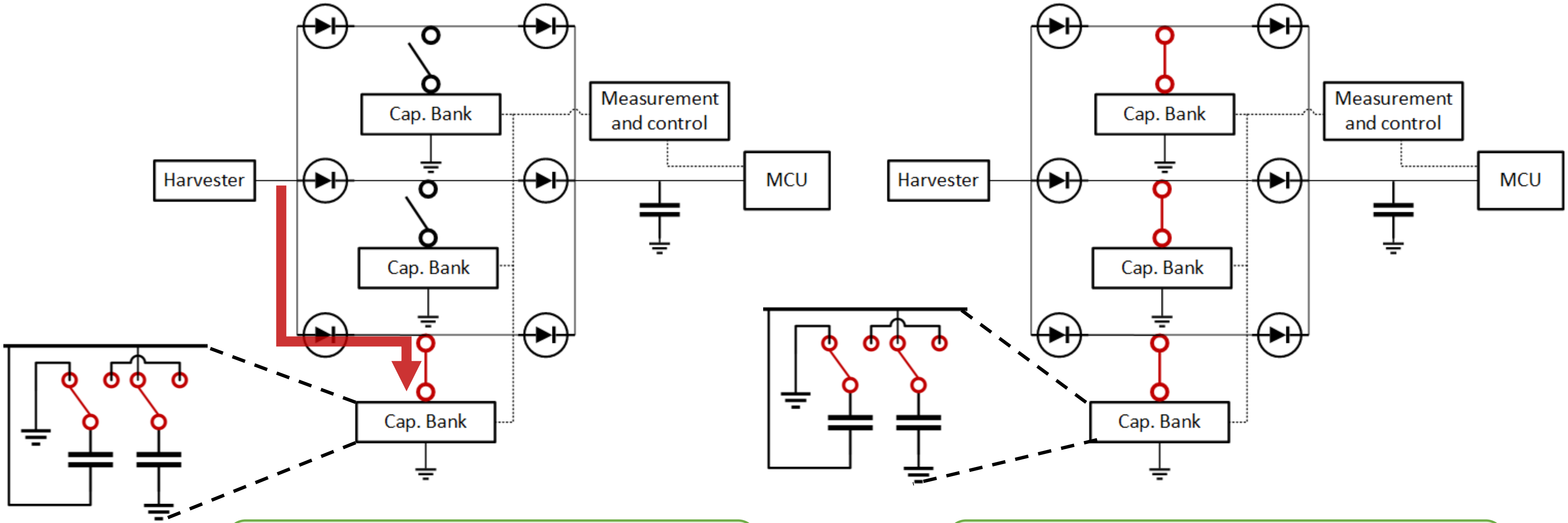
Charge/discharge behavior



Cold start: all energy to small static capacitor

Banks connected in series configuration

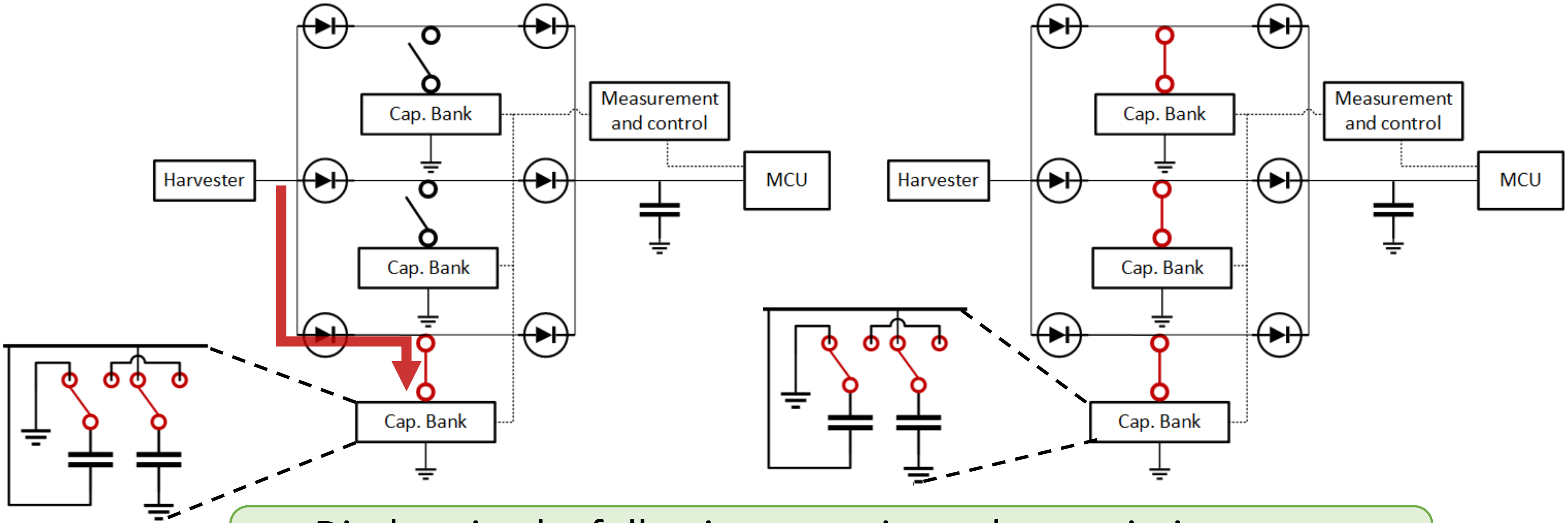
Charge/discharge behavior



Series banks reconfigure to parallel

Additional banks connect in series->parallel

Charge/discharge behavior



Discharging by following opposite order maximizes energy recovered from each bank

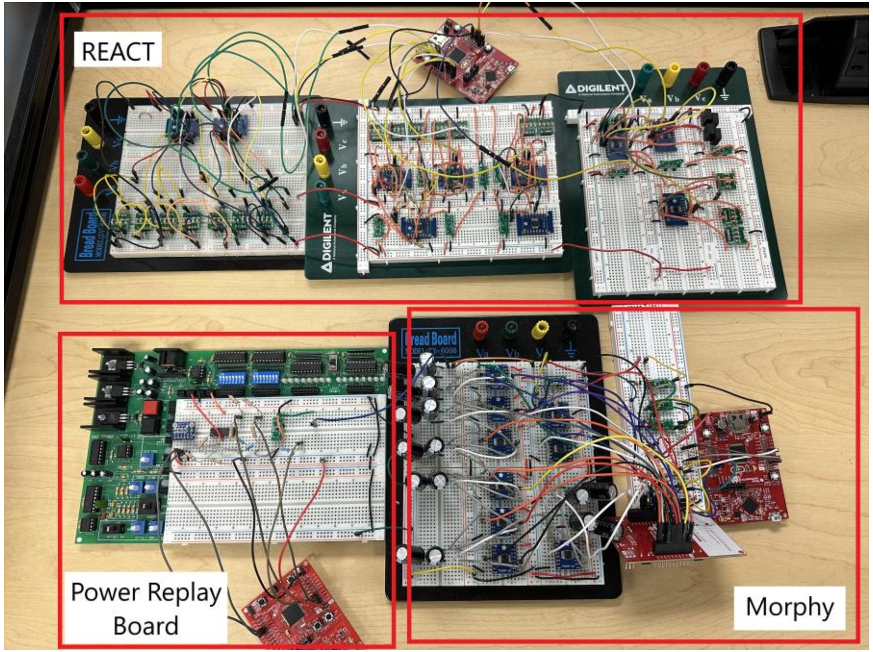
Implementation and Evaluation

Full hardware implementation

- MSP430FR5994 microcontroller
- 4 dynamic banks

Evaluation on real EH traces

- Energy replay system for repeatable experimentation
- Real solar and RF traces



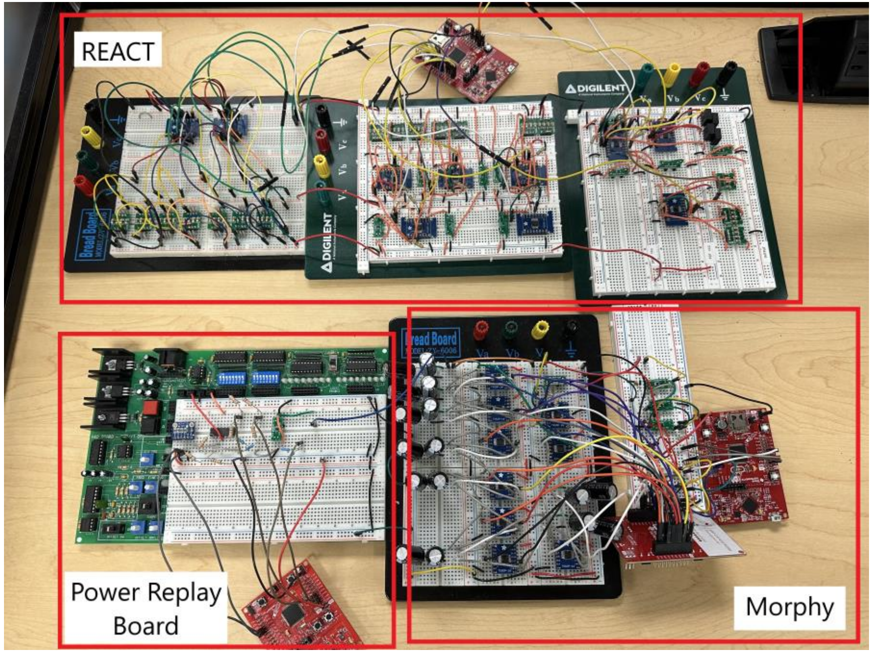
Implementation and Evaluation

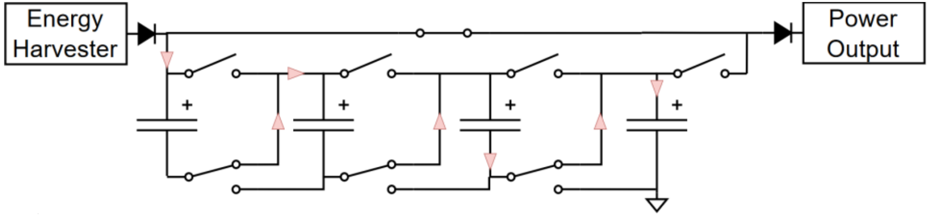
Benchmarks

- Range of reactivity- and capacity-bound workloads
- Sensing, computing, networking

Baseline systems

- 3 single capacitor systems
- Similar work: Morphy





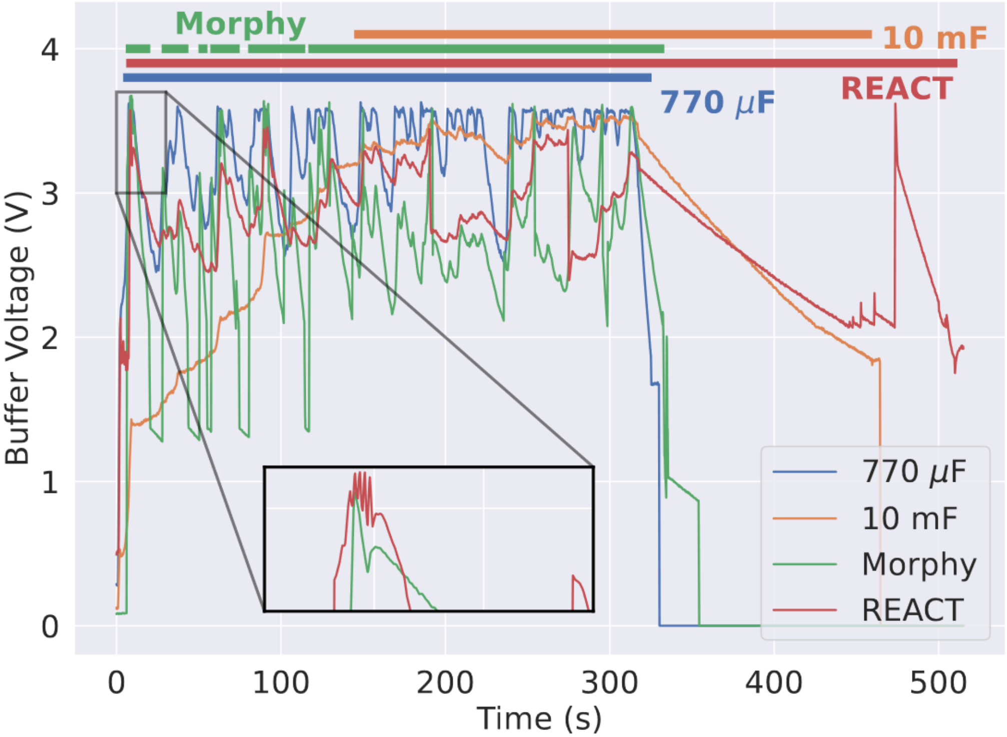
COMPUTER SCIENCE
VIRGINIA TECH.

REACT maximizes system on-time

REACT reaches operational voltage **8x faster** than equivalent-sized static system

REACT runs for **40% longer** than an equally reactive static system

REACT **eliminates switching dissipation** that plagues prior approaches

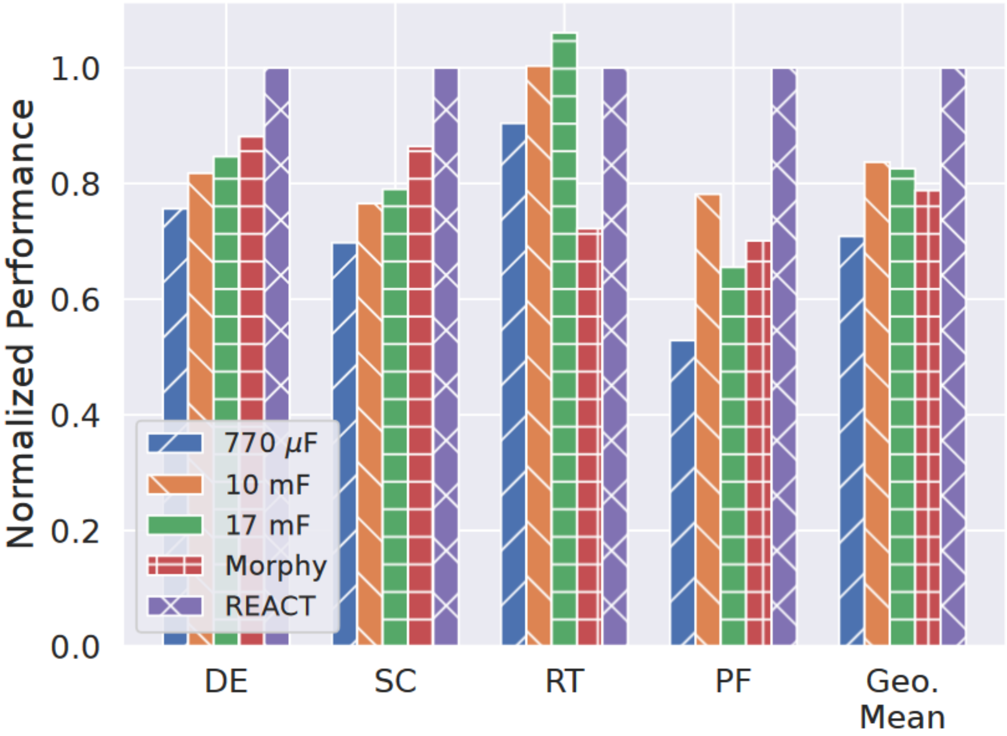


REACT maximizes system on-time

REACT reaches operational voltage **8x faster** than equivalent-sized static system

REACT runs for **40% longer** than an equally reactive static system

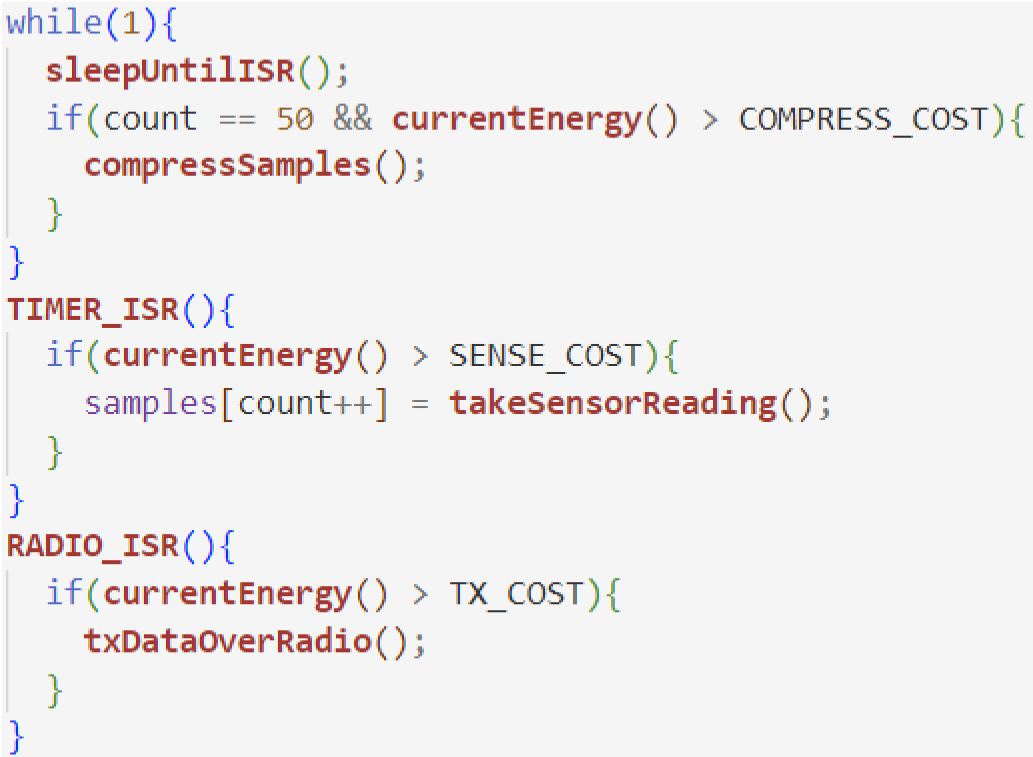
REACT **eliminates switching dissipation** that plagues prior approaches



REACT simplifies batteryless programming

REACT’s **unified, fungible** energy interface simplifies programming

Result: **54% performance improvement** on benchmarks with competing requirements



```
while(1){
  sleepUntilISR();
  if(count == 50 && currentEnergy() > COMPRESS_COST){
    compressSamples();
  }
}
TIMER_ISR(){
  if(currentEnergy() > SENSE_COST){
    samples[count++] = takeSensorReading();
  }
}
RADIO_ISR(){
  if(currentEnergy() > TX_COST){
    txDataOverRadio();
  }
}
```

REACT summary

Batteryless systems face competing performance requirements that render static capacitors insufficient

Prior approaches either fragment energy and complicate system design or face significant energy overheads

REACT **eliminates** the charge time/capacity tradeoff and efficiently combines energy in a unified variable-capacitance buffer

Result: intuitive programming interface and performant intermittent execution **decoupled** from the power frontend

https://github.com/FoRTE-Research/REACT-Artifact