Journey from Mobile Platforms to Self-Powered Wearable Systems

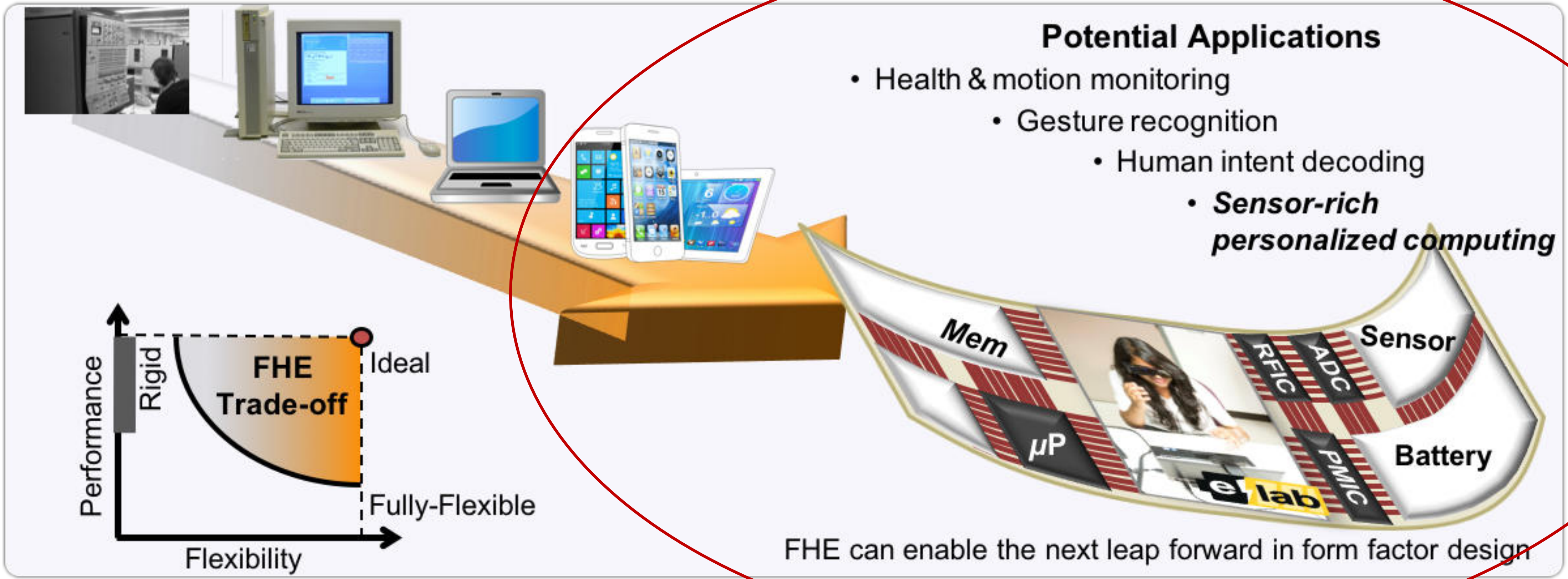
Umit Y. Ogras

ECE 610 – Seminar in Electrical and Computer Engineering

30 October 2020

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Systems on-*Chip* to *Systems-on-Polymer*



- **Impressive progress, but we still need to**
  – Carry a bulky device, re-charge everyday, rely on primitive interaction, …
- **Towards self-powered mobile & wearable systems** that can **understand the user**

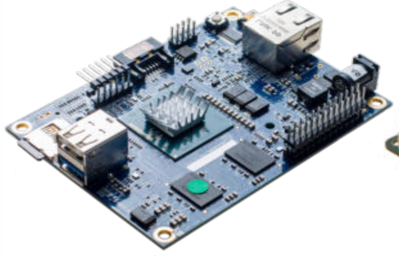
Outline

- **Dynamic Management of Mobile Platforms**
  - Power – temperature dynamics
  - Dynamic Management of Domain-Specific Systems-on-Chip (DSSoC)

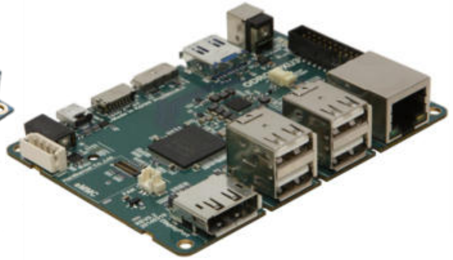
DARPA DASH

Qualcomm Snapdragon

Intel Baytrail



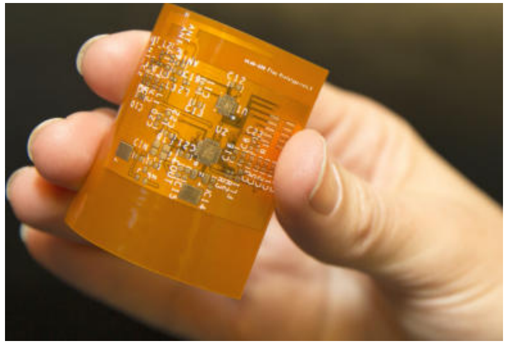
Samsung Exynos

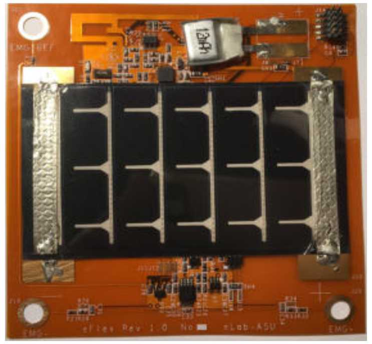


Intel Core i5

Huawei P8

- **Wearable Systems-on-Polymer using Flexible Hybrid Electronics**
  - Flexibility-aware design
  - Optimal energy harvesting
  - Health & activity monitoring





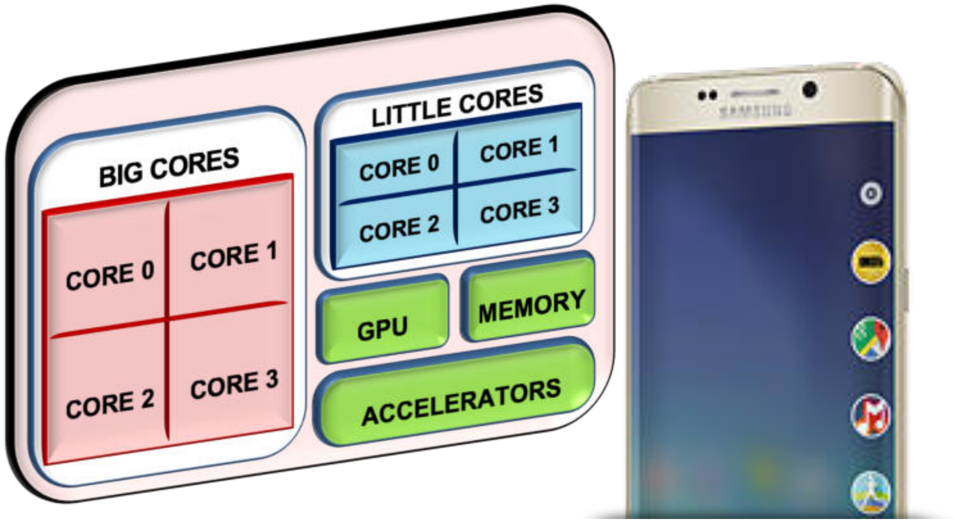
Custom prototypes

Heterogeneity in Computing

- **Heterogeneity is pervasive from low-cost wearable devices to high-end SoCs**
  – Multiple (big/little) cores, vector processing unit, GPU, video, audio, security
- **The design complexity grows faster than our ability to manage**
- **Larger number of cores and knobs lead to intricate dependencies**
  – Different resources, such as, CPU, GPU or memory, become bottleneck as a function of active applications

Applications

Operating System



Hardware / Firmware

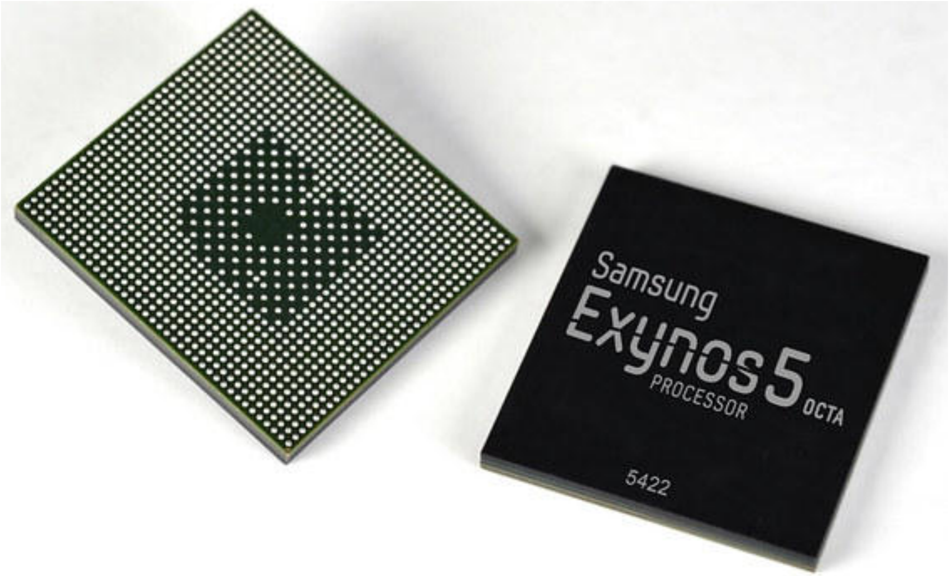
Heterogeneous SoCs

Heterogeneous Architectures

ARM LITTLE CPU	ARM LITTLE CPU	ARM big CPU	ARM big CPU
ARM LITTLE CPU	ARM LITTLE CPU	ARM big CPU	ARM big CPU

Low-power

High-performance



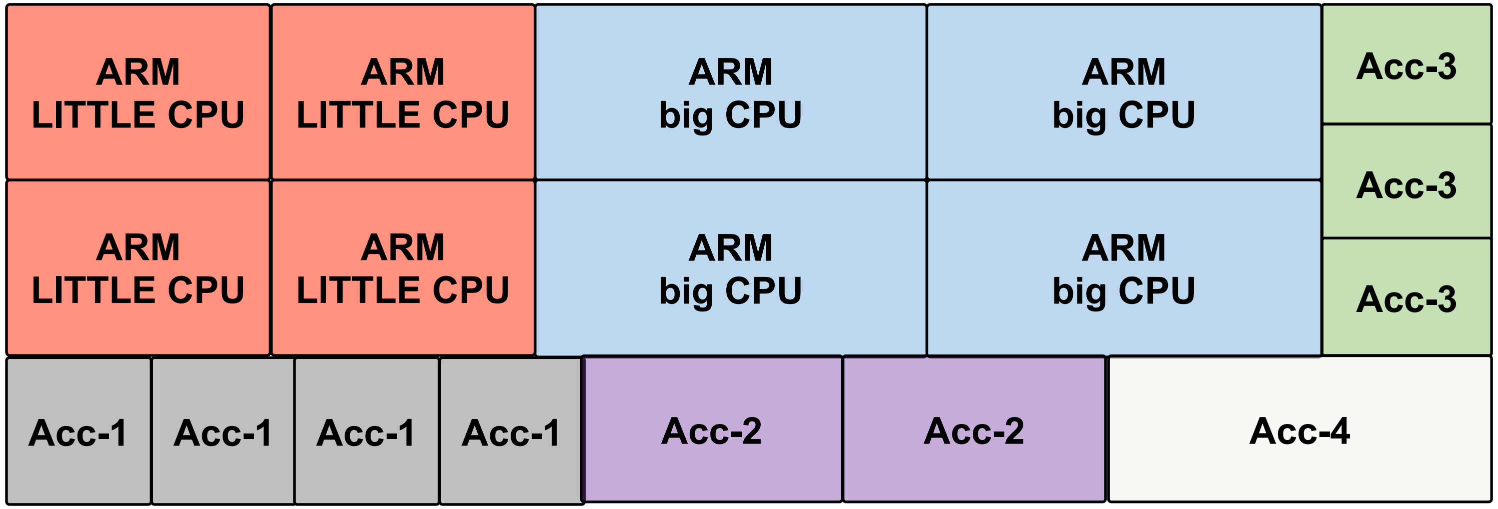
✓ Better match execution resources with application needs

✓ Improved performance, energy-efficiency

✗ Significant gap with respect to special-purpose solutions

Domain-Specific SoCs

Domain-Specific Systems-on-Chip (DSSoC)

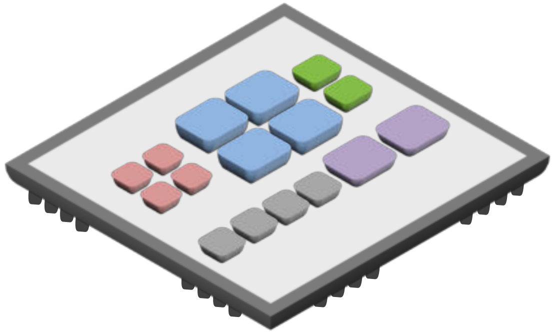



Low-power


High-performance


Specialized processing

- ✓ Judiciously combine general-purpose, special-purpose and hardware accelerator cores
- ✓ Highly efficient for domain applications
- ✓ Flexibility to execute other domains

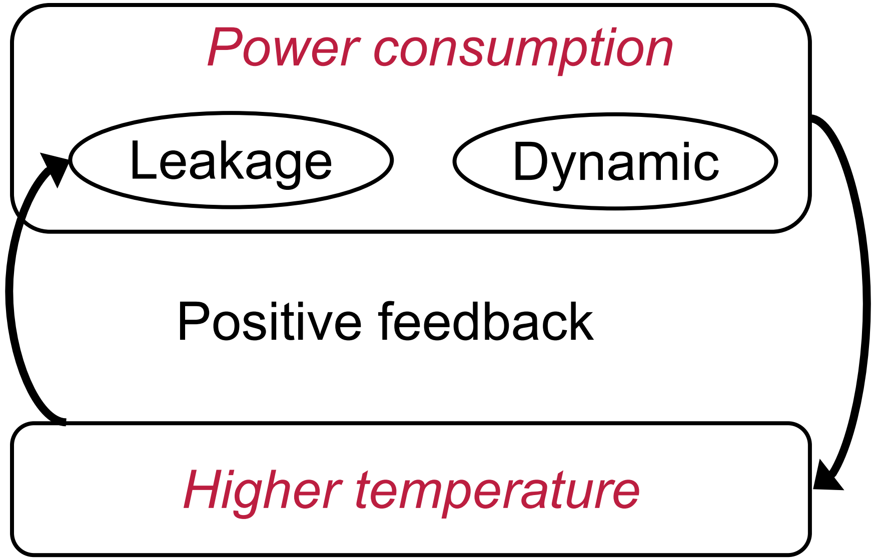


Examples of HW Accelerators:

FFT, matrix multiplier, correlators, encoder/decoder, video processor …

Dynamic Thermal and Power Management for SoCs

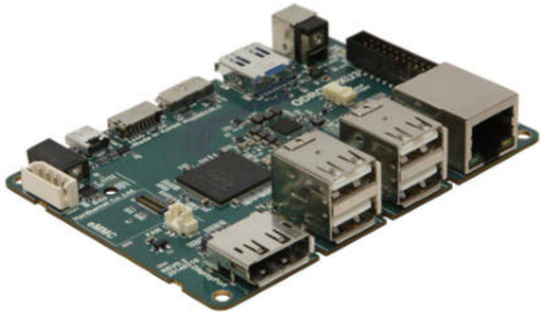
- **Power consumption and heat dissipation are major problems for mobile platforms**
  - High temperature affects user experience and reliability
  - Power and temperature form a positive feedback system

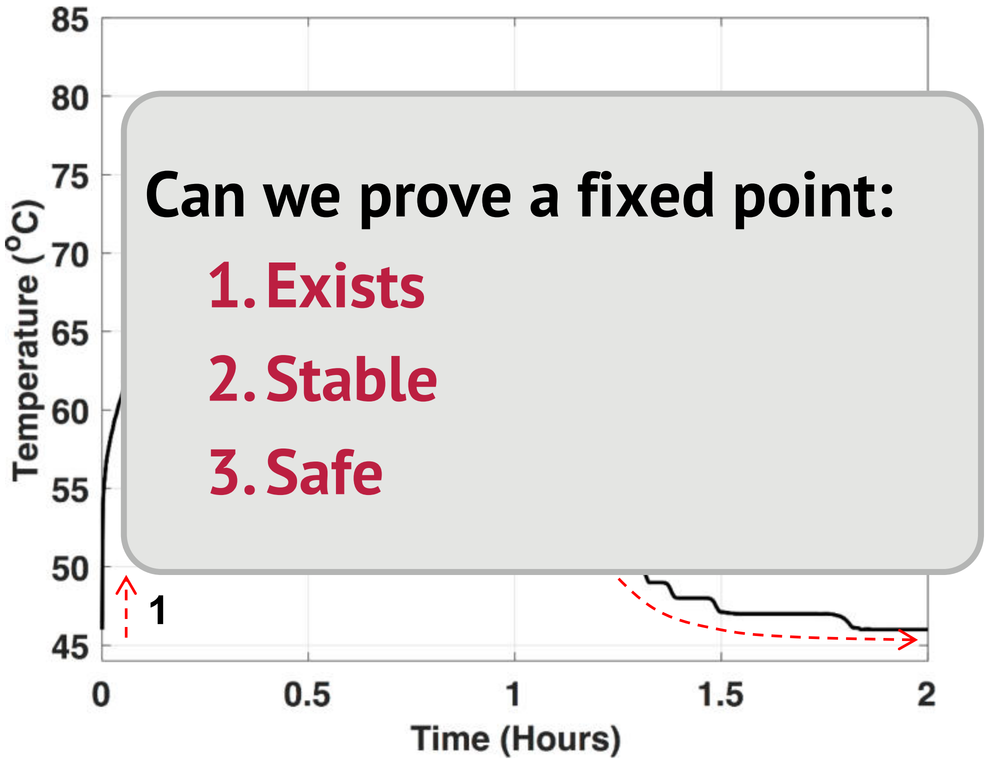


- **Theoretically grounded** and **practical techniques**
  1. **Power – temperature stability analysis**
  2. **Online learning for GPU performance modeling and management**

Illustration: Power – Temperature Dynamics

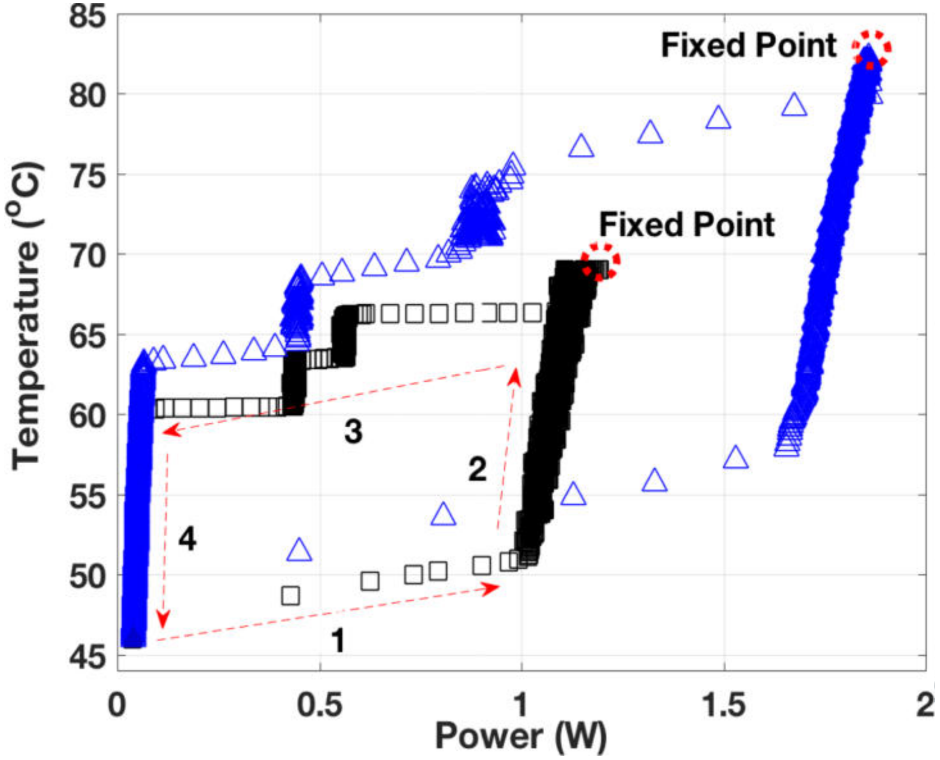
- A complete heat-up/cool-down cycle on Exynos 5422 SoC
- With larger dynamic activity





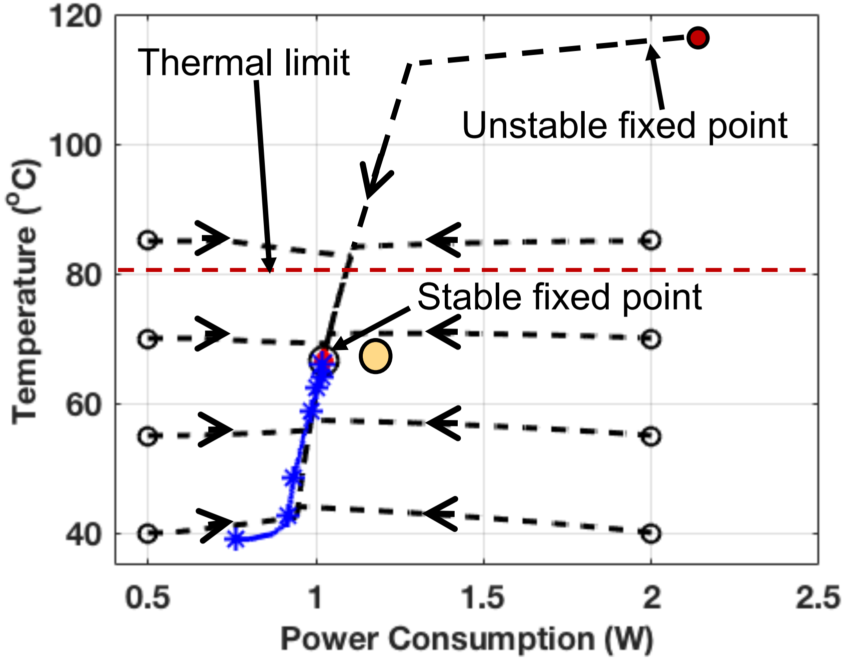
Can we prove a fixed point:

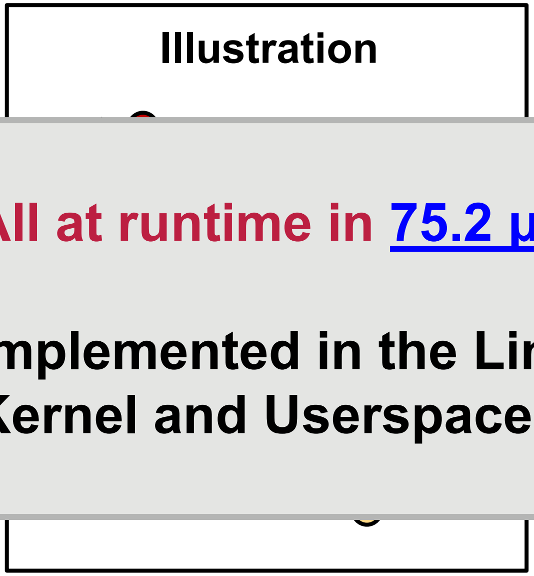
1. Exists
2. Stable
3. Safe



Before diving into details ....

- **We determine the existence, stability and safety of the fixed points all at runtime**





All at runtime in [75.2 μs]()

Implemented in the Linux Kernel and Userspace

1. **Derive** the necessary and sufficient conditions for their existence
2. **Prove** the stability of the fixed point(s) and region of convergence
3. **Find** the maximum safe dynamic power consumption
4. **Validate** our approach empirically on 8-core big.LITTLE platform

Thermal and Power Models

- **Temperature at a future time step is given as:**
  - A: Thermal Capacitance
  - B: Thermal Conductance

$$\begin{bmatrix} T_1[k+1] \\ T_2[k+1] \\ \vdots \\ T_N[k+1] \end{bmatrix}_{N\times 1} = A_{N\times N} \begin{bmatrix} T_1[k] \\ T_2[k] \\ \vdots \\ T_N[k] \end{bmatrix}_{N\times 1} + B_{N\times M} \begin{bmatrix} P_1[k] \\ P_2[k] \\ \vdots \\ P_M[k] \end{bmatrix}_{M\times 1}$$

- **Power consumption is the sum of dynamic and temperature dependent leakage power**

$$P_i = \boxed{C_{sw,i} V_i^2 f_i + I_{g,i} V_i} + V_i \kappa_{1,i} T_i^2 e^{\frac{\kappa_{2,i}}{T_i}}$$

$$P_i = \boldsymbol{P_{C,i}} + V_i \kappa_{1,i} T_i^2 e^{\frac{\kappa_{2,i}}{T_i}}$$

- **Nonlinear MIMO system**
  - No closed-form solutions for this nonlinear system
- **Iterative approaches do not work**
  - Convergence of the system is unknown
  - Recursive iterations take a long time to complete
- **Proposed approach**
  - Reduce to SISO model to find an initial estimate
  - Use Newton's method to solve the MIMO system

$C_{sw,i}$: Swi

$I_{g,i}$: Gate

$\kappa_{1,i}, \kappa_{2,i}$ ar ... resource

Fixed Point Function on the SISO Model

- Thermal safety is determined by the maximum temperature T $\quad T = \max_{1 \le i \le N} T_i[k]$
- At steady state ($k \rightarrow \infty$) we can model each hotspot as, where a and b are found using system identification $\quad T = aT + bP$
- Substitute the power model $\quad (1-a)T - bP_C = bV\kappa_1 T^2 e^{\frac{\kappa_2}{T}}$
- Introduce change of variables $\quad \tilde{T} = -\frac{\kappa_2}{T},\ \alpha = \frac{bP_C}{(a-1)\kappa_2} > 0,\ \beta = \frac{(a-1)}{bV\kappa_1\kappa_2}$
- With the new variables, we can write $\quad \boxed{\beta\tilde{T}\left(1-\alpha\tilde{T}\right) = e^{-\tilde{T}}}$
- Obtain the fixed point function by taking logarithm on both sides $\quad \boxed{\boldsymbol{\mathcal{F}\left(\tilde{T}\right) = \ln\beta + \ln\tilde{T} + \ln(1-\alpha\tilde{T}) + \tilde{T}}}$

If $\mathcal{F}\left(\tilde{T}\right) = 0$, then there is a fixed point!

Existence and Stability of Fixed Points

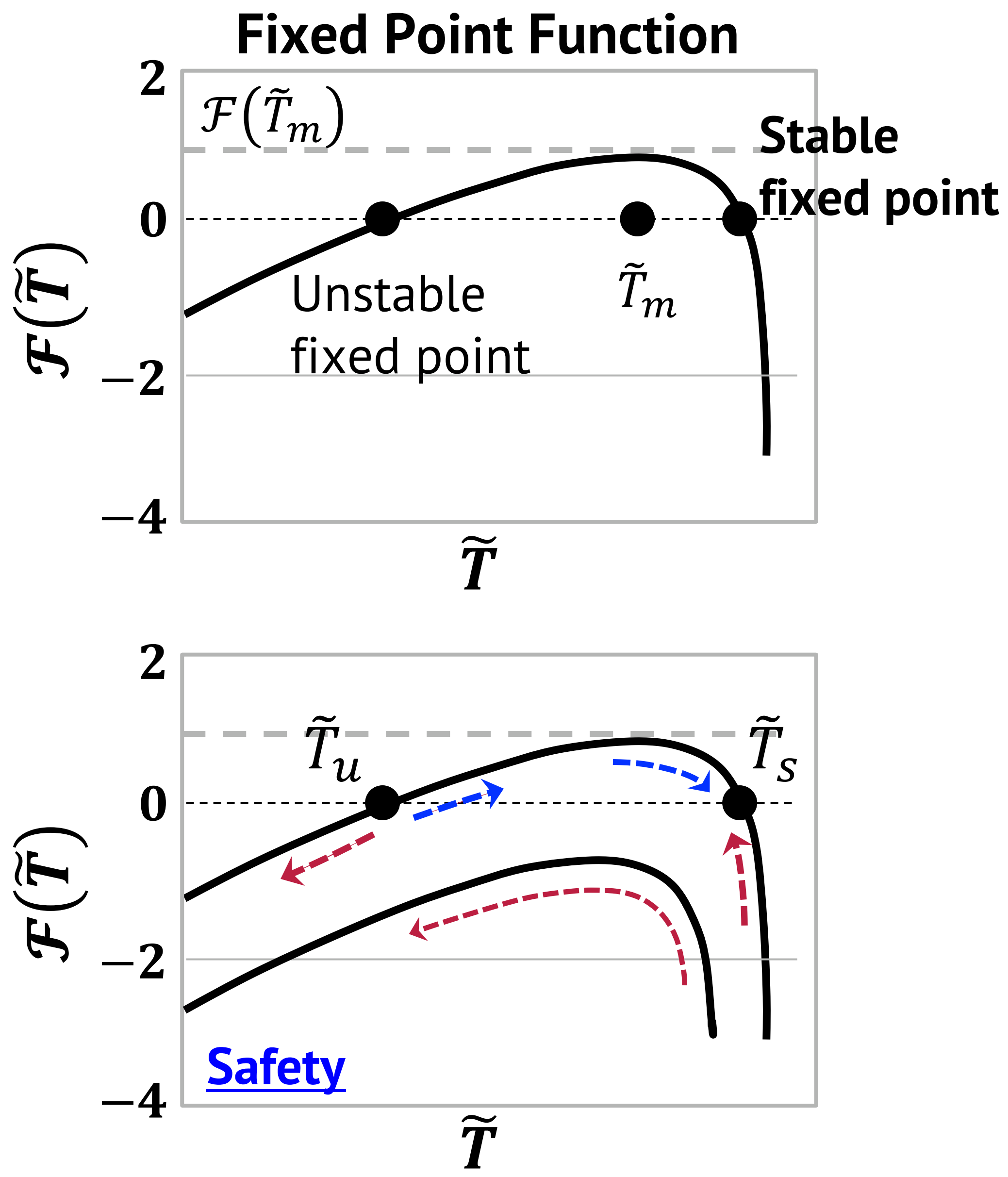
Lemma 1: $\mathcal{F}(\tilde{T})$ **satisfies the following properties**

- $\mathcal{F}(\tilde{T})$ is concave in the interval $\tilde{T} \in (0, \frac{1}{\alpha})$
- $\mathcal{F}(\tilde{T})$ has a unique maxima at $\tilde{T}_m$

Theorem 1: $\mathcal{F}(\tilde{T})$ has two fixed points if and only if $\beta \geq \left(\frac{2}{\tilde{T}_m} + 1\right) e^{-\tilde{T}_m}$

Theorem 2: Stability of fixed points

- When $\mathcal{F}(\tilde{T})$ has no solution, the temperature iteration diverges, i.e, $\tilde{T} \to 0$ $(T \to \infty)$
- Where there are two fixed points, $\tilde{T}_u \in (0, \tilde{T}_m)$ is unstable and $\tilde{T}_s \in (\tilde{T}_m, 1/\alpha)$ is stable



Extension to the MIMO Solution

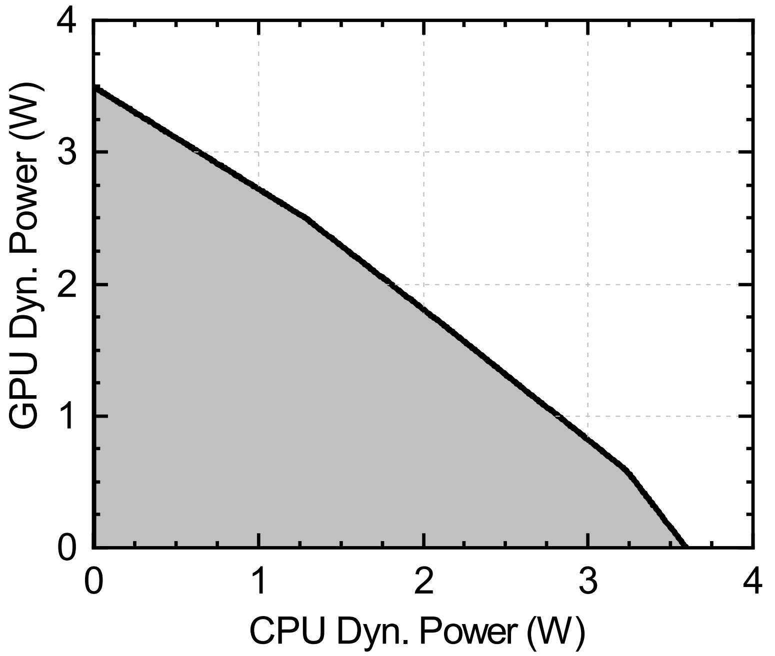
- **The fixed point function can be written as**

$$\mathbf{f}(T_1, \dots, T_N) = A\mathbf{T} + B[P_1, P_2, \dots, P_M]^T - \mathbf{T} = 0$$

- **Find the SISO solution to find the initial solution**
- **Employ Newton's method using the SISO solution as the initial point**

- **Found the region where the convergence is guaranteed**
- ***Matches with experimental results***

Convergence in less than five iterations

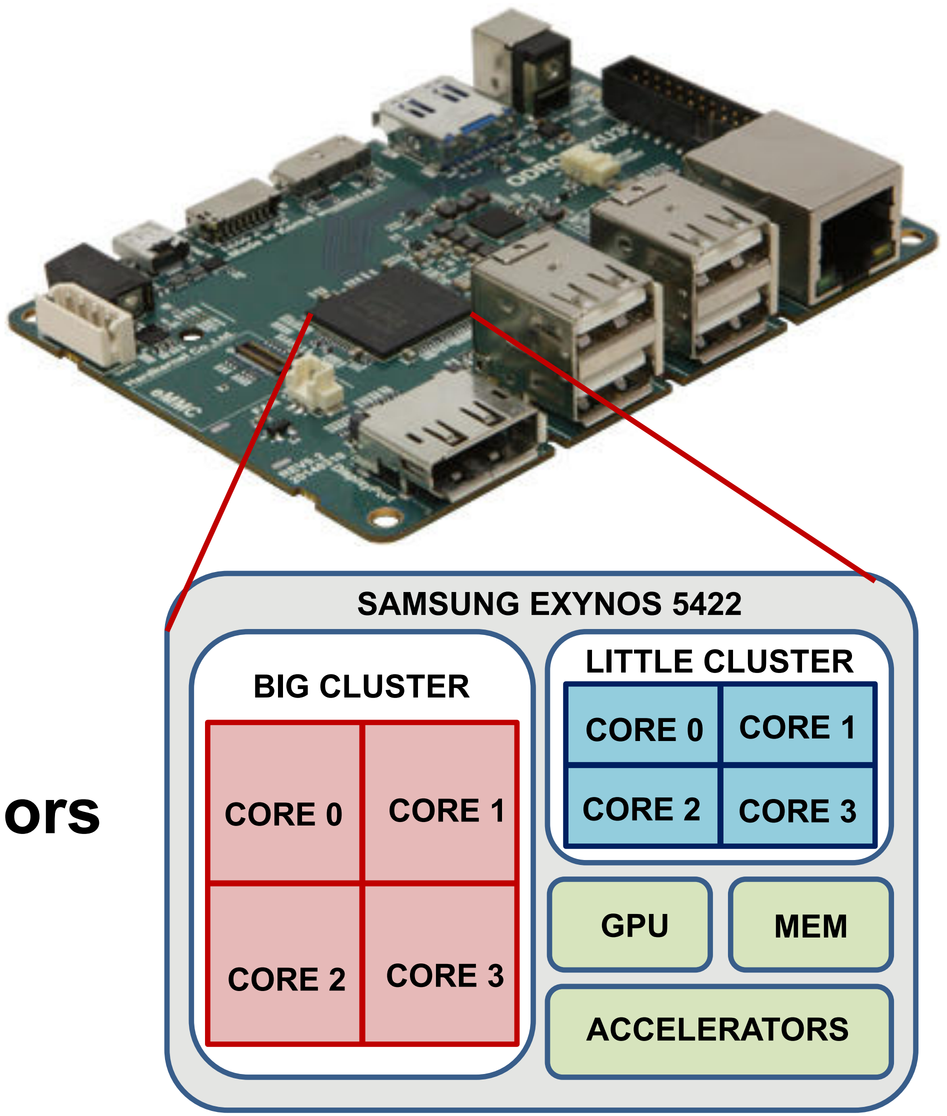


G. Bhat, S. Gumussoy, and U. Y. Ogras, "Power-Temperature Stability and Safety Analysis for Multiprocessor Systems," *ACM Tran. on Embedded Comp. Sys. (ESWEEK Special Issue)*, October 2017

G. Bhat, S. Gumussoy, and U. Y. Ogras. "Analysis and Control of Power-Temperature Dynamics in Heterogeneous Multiprocessors." *IEEE Transactions on Control Systems Technology* (preprint 2020)

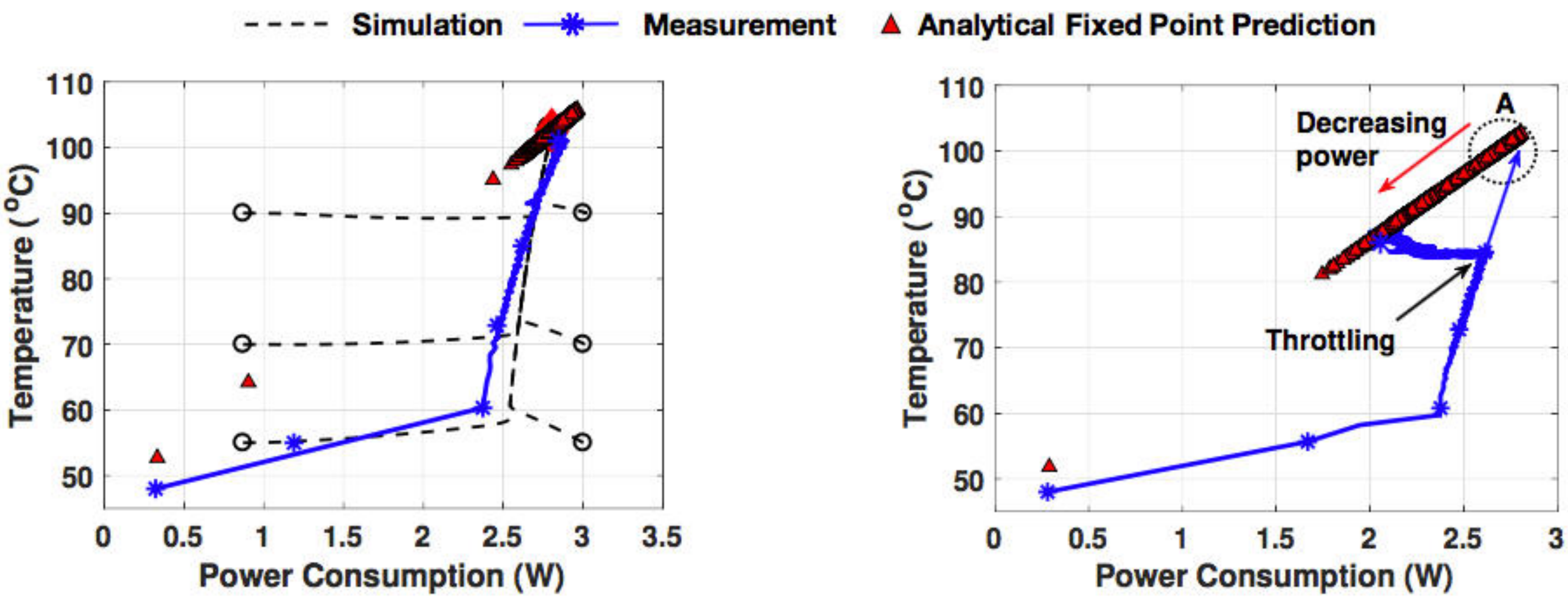
Experimental Setup

- **Odroid-XU3 board**
  - Exynos 5422 Octa-core CPU
  - 4 little and 4 big cores
- **Validated using 10 different apps running on Android 4.4.4:**
  - MiBench
  - Parsec
  - SPEC
- **Invoked every 100 ms with the default governors**
- **Our technique takes 75.2 µs**



Time Granularity of Our Predictions

- *Fixed point predictions* match with *measurement* and *simulation*



- **Predictions quickly adapts to change in power consumption,** *including thermal throttling*

Summary of Results

Benchmark	Avg. Total Power (W)	Avg. Dyn. Power (W)	Empirical Fixed Point (℃)	Comput. Fixed Point (℃)	Abs. Pred. Error (℃)	Experiment Duration (s)
Idle @ 1.3 GHz	0.38	0.31	51.8	52.2	0.4	3979
Idle @ 1.5 GHz	0.47	0.38	54.4	55.5	1.1	4045
Idle @ 1.8 GHz	0.70	0.59	60.2	60.1	0.1	4171
Idle @ 2.0 GHz	1.01	0.87	66.0	66.6	0.6	3413
Vortex	1.73	1.55	80.0	81.4	1.4	1989
Matrix Mult.	1.84	1.65	83.0	83.8	0.8	521
CRC32	2.04	1.83	85.0	88.5	3.5	907
Patricia	2.20	1.97	89.0	91.8	2.8	900
Blackscholes	2.42	2.17	94.0	96.6	2.6	785
Streamcluster	2.48	2.22	94.0	97.9	3.9	614
Basicmath	2.49	2.24	93.0	98.3	5.3	492
Fluidanimate	2.50	2.26	93.0	98.8	5.8	550
FFT	2.71	2.41	101.0	102.5	1.5	729
Streamcluster+CRC32	3.14	2.79	109.0	111.5	2.5	1132

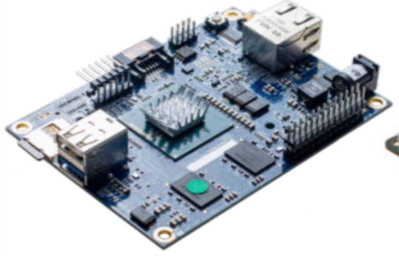
Outline

- **Dynamic Management of Mobile Platforms**
  - Power – temperature dynamics
  - **Dynamic Management of Domain-Specific Systems-on-Chip (DSSoC)**

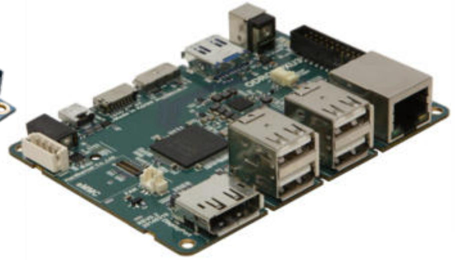
DARPA DASH

Qualcomm Snapdragon

Intel Baytrail



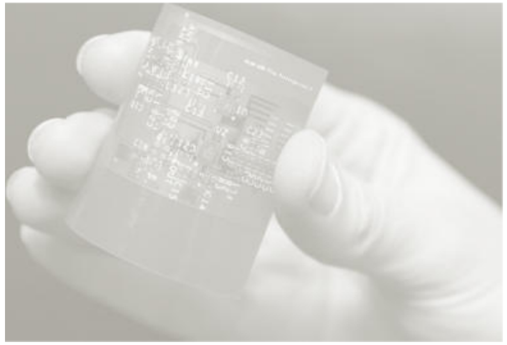
Samsung Exynos

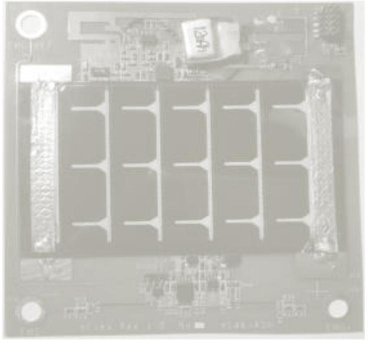


Intel Core i5

Huawei P8

- **Wearable Systems-on-Polymer using Flexible Hybrid Electronics**
  - Flexibility-aware design
  - Optimal energy harvesting
  - Health & activity monitoring

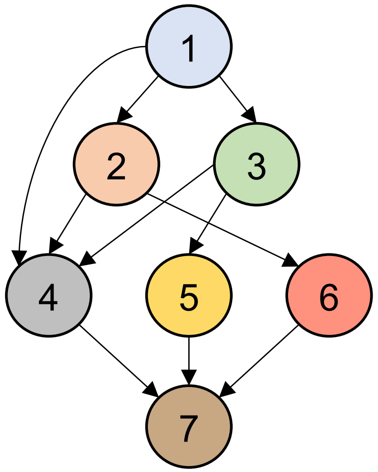




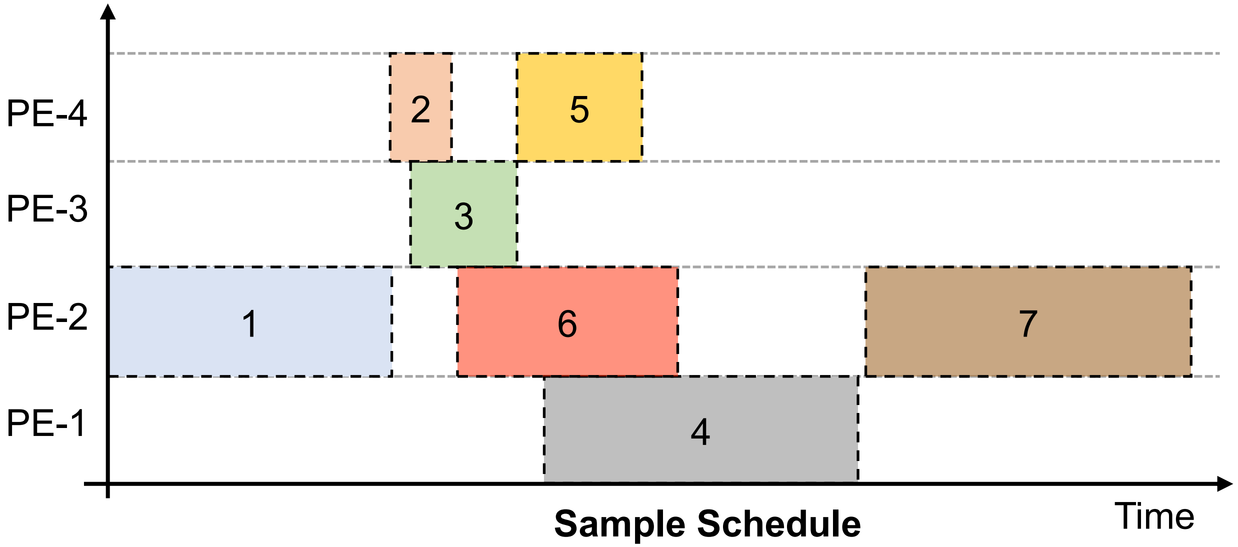
Custom prototypes

Harvest Full Potential of DSSoCs-1: Scheduling

- **How to harvest the full potential of DSSoCs?**
  – Optimally utilize the processing elements (PEs) at runtime
  – Make acceleration of domain-specific applications oblivious to developers
- **Task scheduling:**
  – Assign tasks to PEs to achieve optimization goals
    - Minimize execution time, power dissipation, energy consumption
  – Applications modeled as directed acyclic graphs (DAGs)



Application DAG

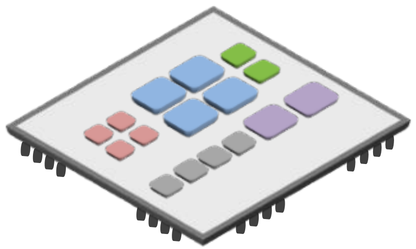


Sample Schedule

S. E. Arda *et al.*, "DS3: A System-Level Domain-Specific System-on-Chip Simulation Framework," in *IEEE Trans. on Computers*, vol. 69, no. 8, pp. 1248-1262, 1 Aug. 2020, doi: 10.1109/TC.2020.2986963.

Challenges of Runtime Scheduling for DSSoC

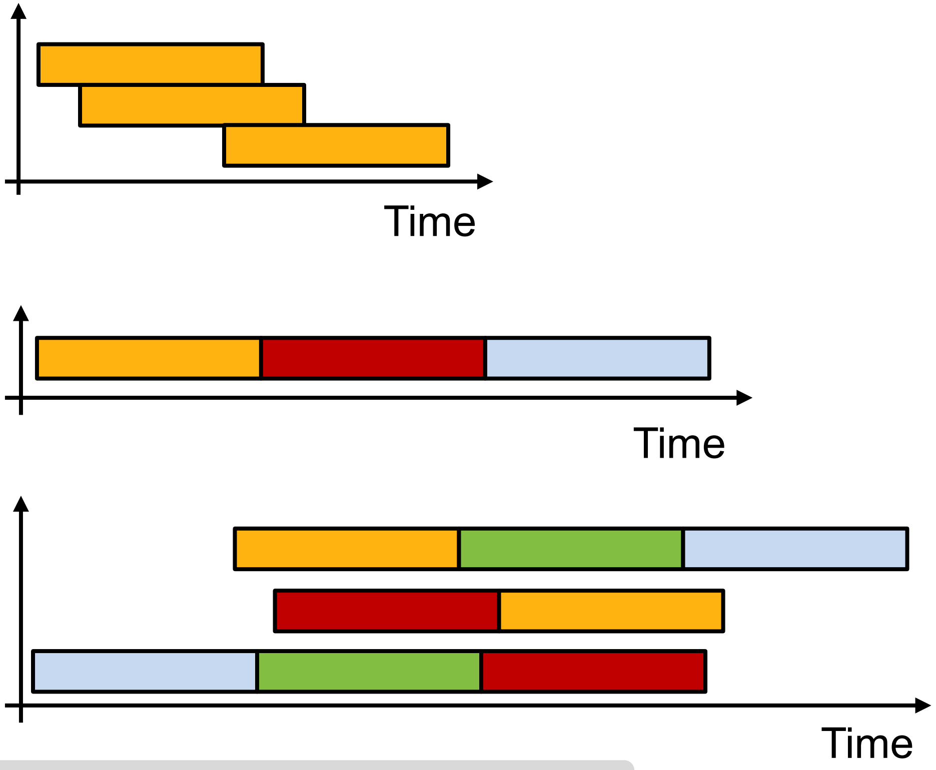
1. **Heterogeneity of the platform:**
   – Many scheduling choices at runtime
   – NP-complete[1][2]



2. **Streaming applications:**
   – Real-time application arrival
   – Application overlap, different incoming system states

3. **Various applications:**
   – Different task and DAG characteristics

4. **Simultaneous execution of multiple applications:**
   – Different types of applications executing simultaneously



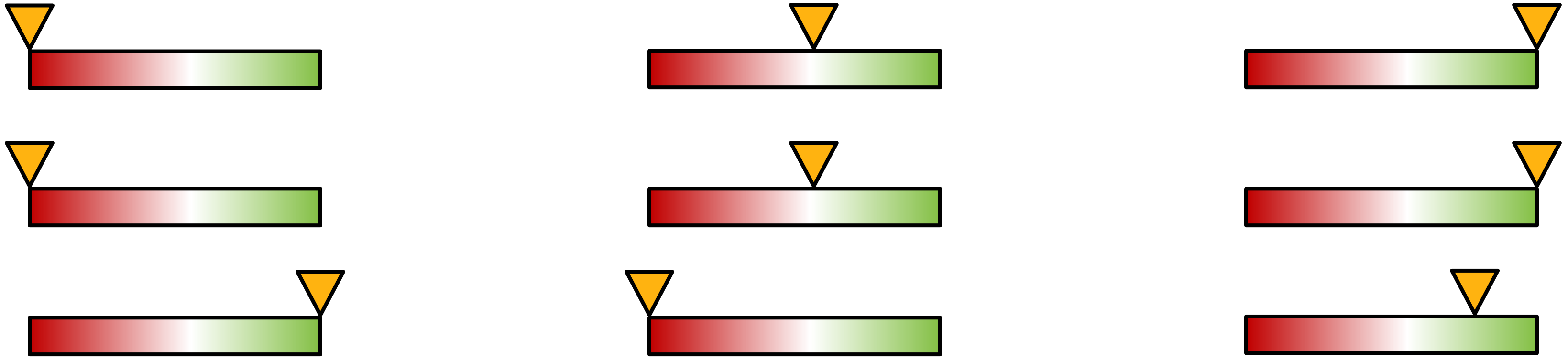
Goal: ***Perform task scheduling for multiple simultaneous streaming applications in heterogeneous many-core platforms***

[1] Gary, Michael R., and David S. Johnson. "Computers and Intractability: A Guide to the Theory of NP-completeness." (1979).
[2] Ullman, Jeffrey D. "NP-complete Scheduling Problems." *Journal of Computer and System Sciences* 10.3 (1975): 384-393.

Scheduling Approaches and Goals

MIP – Mixed Integer Programming
CP – Constraint Programming

Approach / Metric	Optimization (MIP / CP)	Heuristics	Learning-based
Complexity			
Runtime			
Optimality			
	Rossi et. al, 2006	Arabnejad et. al, 2013 Bittencourt et. al, 2010 Topcuglu et. al, 2002 Swaminathan et. al, 2001	Mao et. al, 2016 Mao et. al, 2019 ***Proposed IL-based approach, 2020***



Can we achieve near-optimal results at runtime with a similar cost to a heuristic?

Key Insights & Imitation Learning (IL)-based Scheduling

1. ***Use optimal algorithms offline without being limited by runtime overheads***

2. ***Design a policy that approximates the Oracle with minimum runtime overhead***

3. ***Exploit the effectiveness of ML to learn from Oracle for any objective***

Key Insights & Imitation Learning (IL)-based Scheduling

1. *Use optimal algorithms offline without being limited by runtime overheads*

2. *Design a policy that approximates the Oracle with minimum runtime overhead*

3. *Exploit the effectiveness of ML to learn from Oracle for any objective*

1. *Collect expert samples and construct Oracle*

Optimization-based (and / or) heuristic schedulers

Key Insights & Imitation Learning (IL)-based Scheduling

1. *Use optimal algorithms offline without being limited by runtime overheads*

2. *Design a policy that approximates the Oracle with minimum runtime overhead*

3. *Exploit the effectiveness of ML to learn from Oracle for any objective*

1. Collect expert samples and construct Oracle

Optimization-based (and / or) heuristic schedulers

2. Generate training data for IL

Encode Oracle decisions with selected features

Features that represent the system state

A. Feature Selection for State Representation

- **Goals of feature selection**
  - Identify the minimal set of features that:
    - Lead to high training accuracy, low storage overhead, minimal runtime overhead
  - Features must describe task, DAG (application), processing elements
  - Represent both design-time and runtime behavior
- **Intuitively analyzed the factors that influence scheduling decisions**
  - Execution times, power consumption of tasks
  - Task predecessor data in DAG
  - Availability of processing elements
  - Communication volumes

Feature Type	Feature Description	Feature Categories
Static ($\mathcal{F}_S$)	ID of task-j in the DAG	Task
	Execution time of a task T_j in PE P_i ($t_{exe}(P_i, T_j)$)	Task PE
	Downward depth of task T_j in the DAG	Task Application
	IDs of predecessor tasks of task T_j	Task Application
	Application ID	Application
	Power consumption of task T_j in PE P_i	Task PE
Dynamic ($\mathcal{F}_D$)	Relative order of task T_j in the ready queue	Task
	Earliest time when PEs in a cluster-c are ready for task execution	PE
	Clusters in which predecessor tasks of task T_j executed	Task
	Communication volume from task T_j to task T_k (v_{jk})	Task

Key Insights & Imitation Learning (IL)-based Scheduling

1. *Use optimal algorithms offline without being limited by runtime overheads*

2. *Design a policy that approximates the Oracle with minimum runtime overhead*

3. *Exploit the effectiveness of ML to learn from Oracle for any objective*

1. Collect expert samples and construct Oracle

Optimization-based (and / or) heuristic schedulers

2. Generate training data for IL

Encode Oracle decisions with selected features

Features that represent the system state

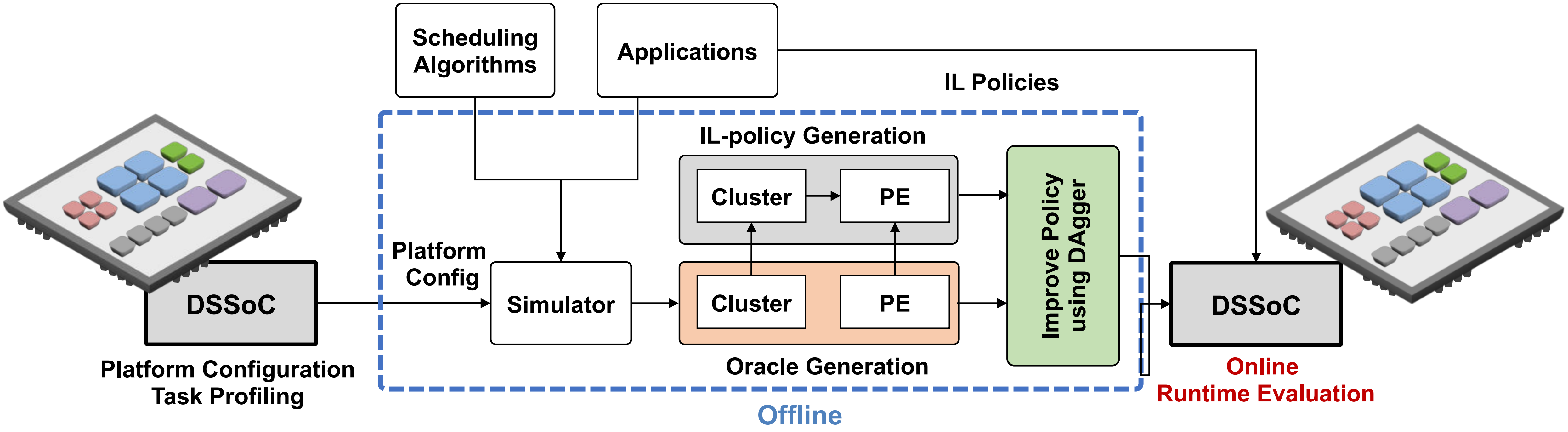
3. Train IL policies and refine using Data Aggregation

Design the IL-scheduler

Supervised learning techniques

Imitation Learning-based Scheduling Framework

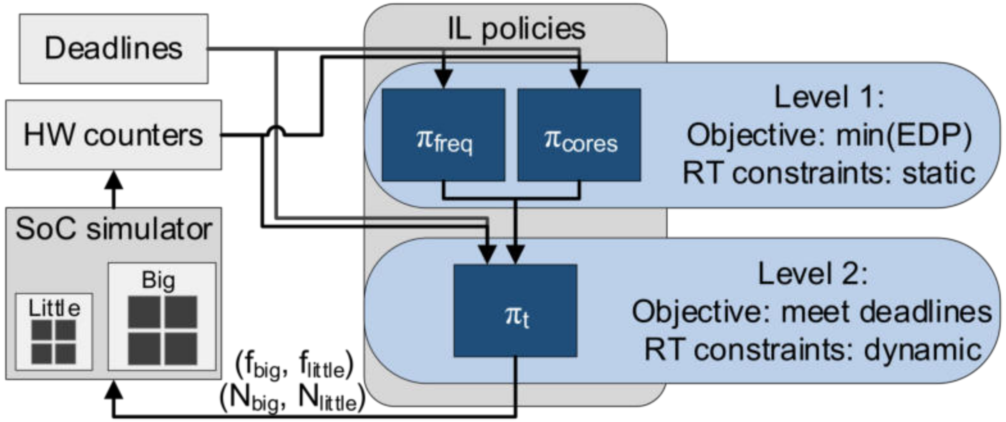
- **A hierarchical IL-based scheduling framework for heterogeneous SoCs**
- **Break down the complex scheduling problem into sub-problems**
  - Group identical PEs into processing clusters
  - First-level IL policy: **Predict cluster**
  - Second-level IL policies: **Predict PEs within the predicted cluster**

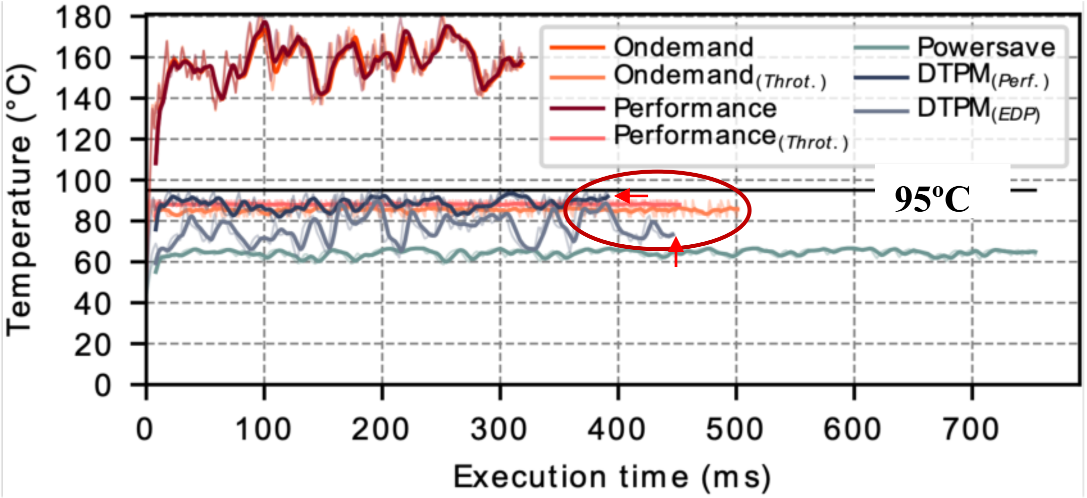


A. Krishnakumar, et al. "Runtime Task Scheduling Using Imitation Learning for Heterogeneous Many-Core Systems," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* 39.11 (2020): 4064-4077.

Harvest Full Potential of DSSoCs-2: Dynamic Power Mgmt

- **Do we have to run all resources all the time?**
  - Selectively turn-off processors if they won't be needed
- **Do active resource need to run at full speed?**
  - Dynamically scale the voltage and frequency to save power/energy
- **Hierarchical Dynamic Thermal-Power Management**
  - **Level-1:** IL policies predict the frequency and number of active cores in each cluster of the SoC
  - **Level-2:** Regression policy predicts the execution time and fine-tunes L1 decisions to meet soft real-time (RT) deadlines
- **Optimization objectives:**
  - Improve energy efficiency
  - Improve performance
  - Real-time aware optimization
  - Thermal-aware optimization





Sartor, Anderson L., et al. "HiLITE: Hierarchical and Lightweight Imitation Learning for Power Management of Embedded SoCs," *IEEE Computer Architecture Letters*, 19.1 (2020): 63-67.

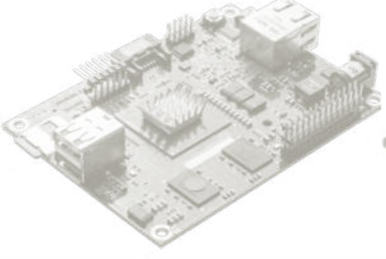
Outline

- Dynamic Management of Mobile Platforms
  - Power – temperature dynamics
  - **Online learning for frequency sensitivity**

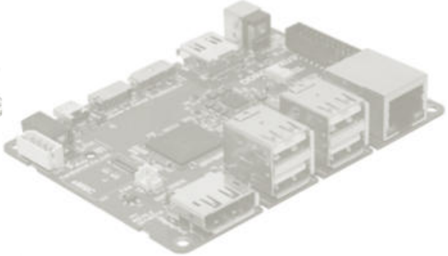
DARPA DASH

Qualcomm Snapdragon

Intel Baytrail



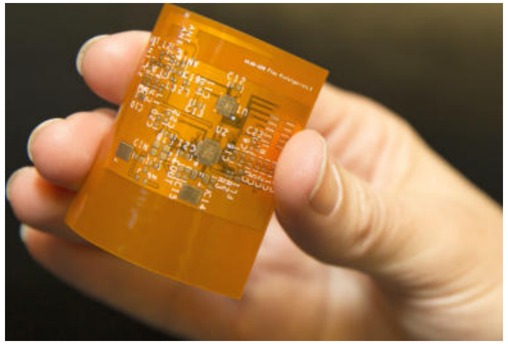
Samsung Exynos

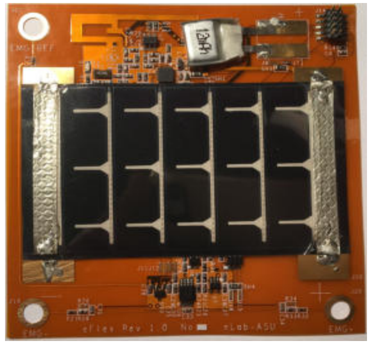


Intel Core i5

Huawei P8

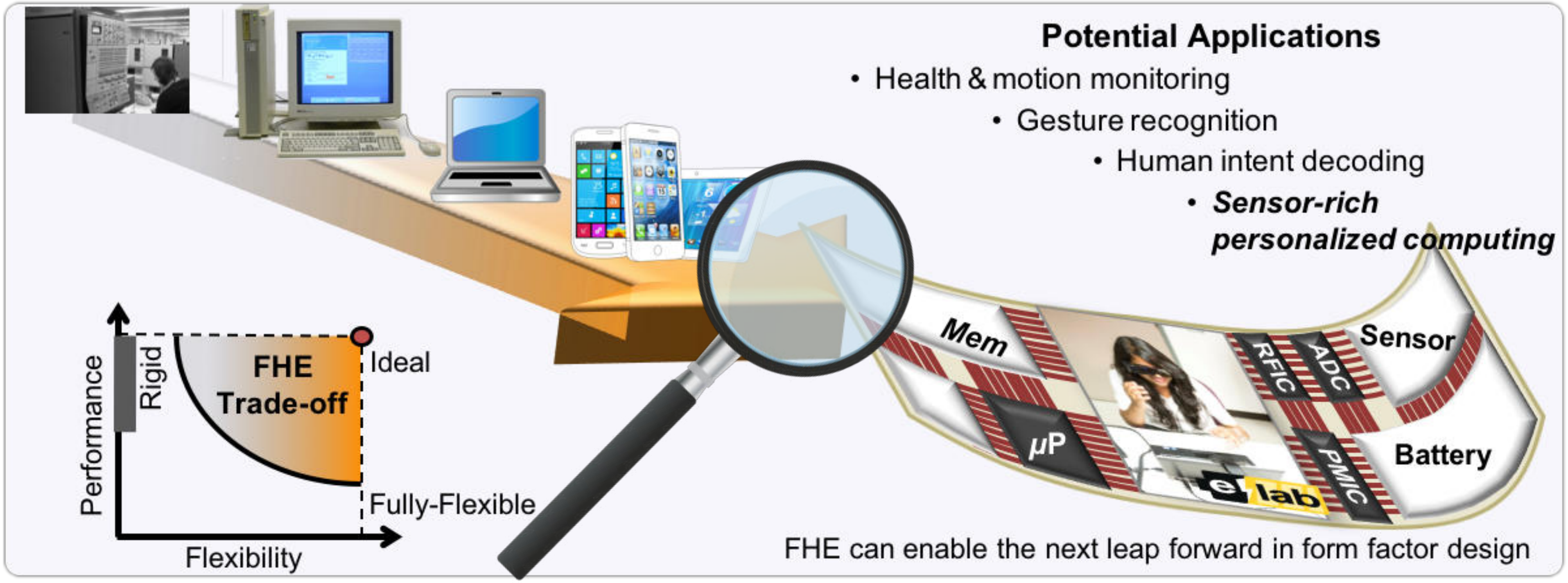
- **Wearable Systems-on-Polymer using Flexible Hybrid Electronics**
  - **Flexibility-aware design**
  - **Optimal energy harvesting**
  - **Health & activity monitoring**





Custom prototypes

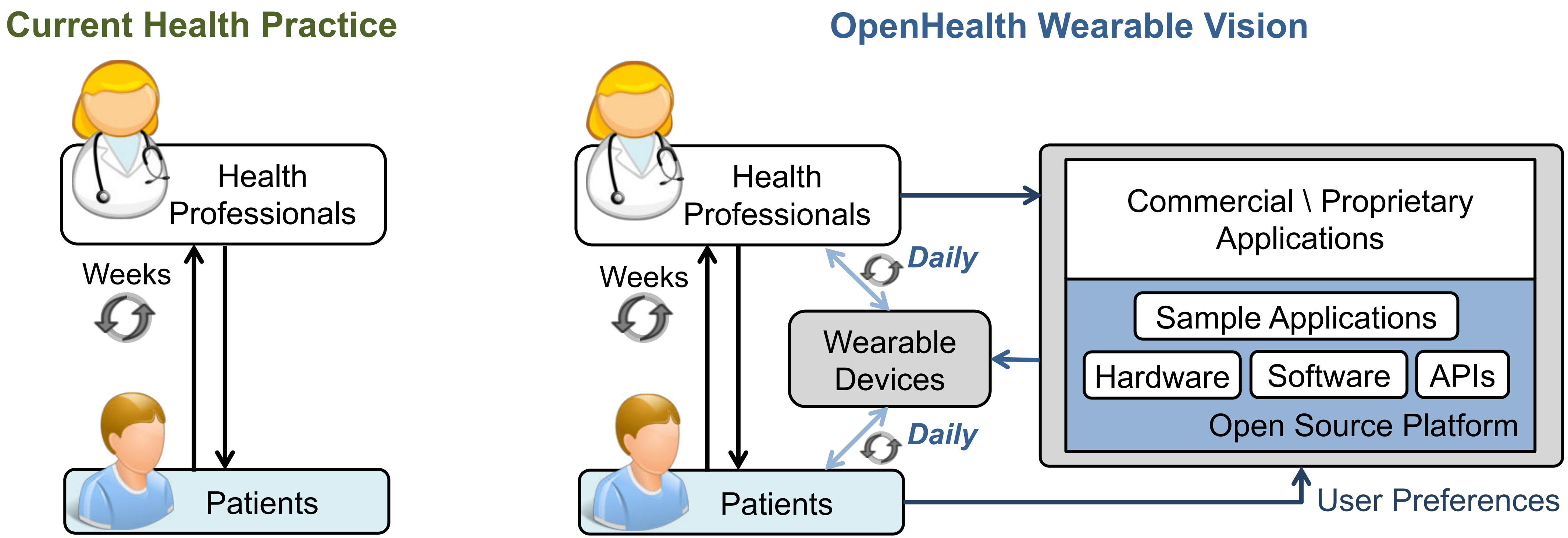
Systems-on-*Chip* to *Systems-on-Polymer*



- Flexible electronics refer to circuits on bendable, rollable, or elastic substrates
- Despite impressive potential, they are significantly larger and slower
  - Successful applications to displays, sensors, and solar cells
- **Emerging FHE target the limitations of flexible electronics**
  - Combine the capabilities of silicon ICs with the physical benefits of flexible electronics

Health Monitoring using Mobile Devices

- **15% of the world's population lives with a disability**
- **110-190 million people face significant difficulties in functioning**
- ***Intl. Parkinson and Movement Disorders Society Task Force on Technology:***
  - Low-cost and small form-factor wearable devices offer great potential
  - Enabled by advances in low power sensors, processors, communications

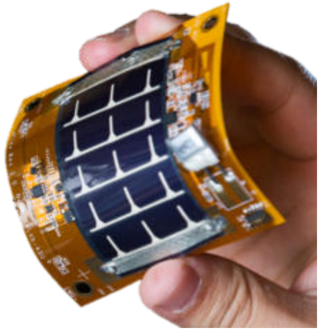


Why Online Learning on Wearable Devices

- **Smartphones have been popular:**
- **But they are not appropriate**
  - *Some patients cannot even carry them*
  - Large power consumption & charging requirements
  - Cannot provide real-time guarantees (e.g., sampling rate)
  - ***They are not designed for this purpose***

- **Existing work on wearables and smartphones**

	Offline	Online
Data Collection	✓	✗ → ✓
Learning	✓	✗ → ✓
Inference	✓	✓

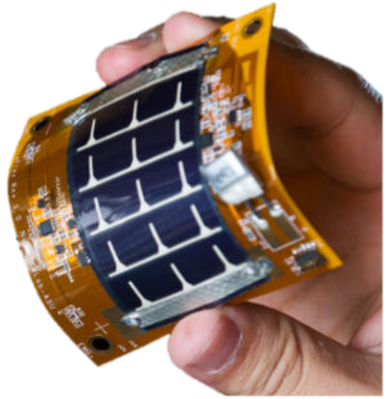


- **Our solution**
  - Physically flexible wearable form-factor
  - Low power & Energy-harvesting
  - Adapt to new users and varying conditions

Our Solution to Wearable Health Monitoring

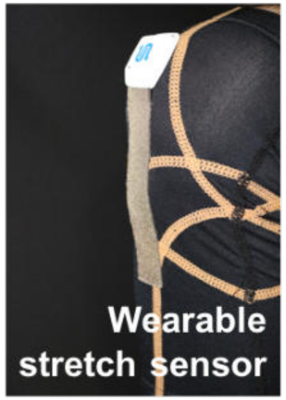
- **We address *adaptation* & *technology* challenges**
  1. **Comfort** → Flexible Hybrid Electronics (FHE)
  2. **Compliance** → Energy Neutral Operation
  3. **Applications** → Movement Disorders

- **Human Activity Recognition (HAR) & Gait Analysis**
  – Patient rehabilitation
  – Fall detection
  – Physical activity promotion

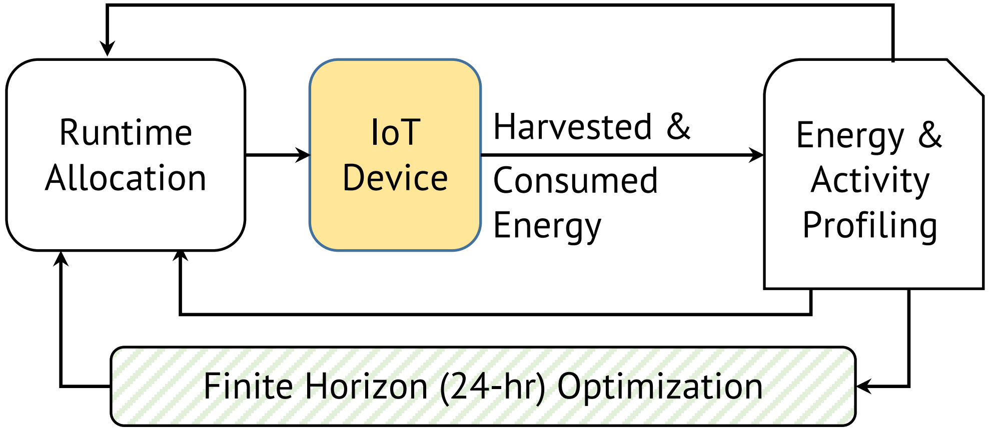


Optimal energy harvesting & management

Flexible and Stretchable Devices

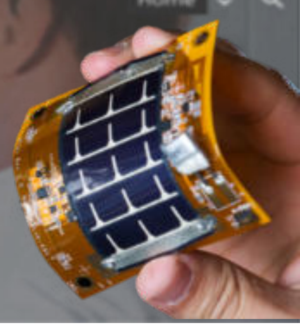


Closed-Loop Optimal Energy Allocation

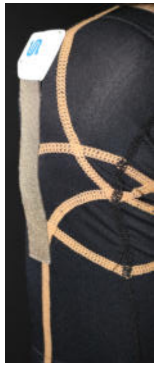


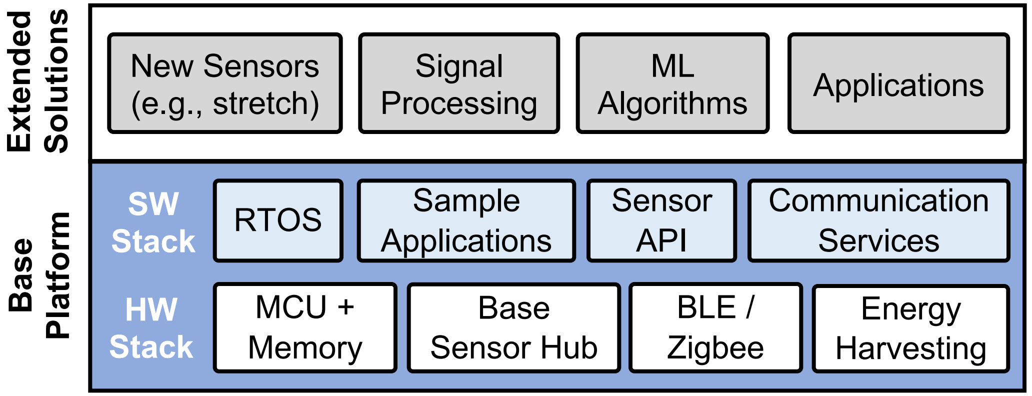
OpenHealth

An open-source HW-SW platform released to public



- **Aim to provide a common compatible HW/SW platform**
  Hardware
  – Ti CC26x2r microcontroller with BLE 5.0 stack
  – 9-axis IMU: MPU-9250 (Accelerometer, gyroscopre, magnetometer)
  – Flexible bending sensor and stretch sensor
  – SD card slot for on-board data backup

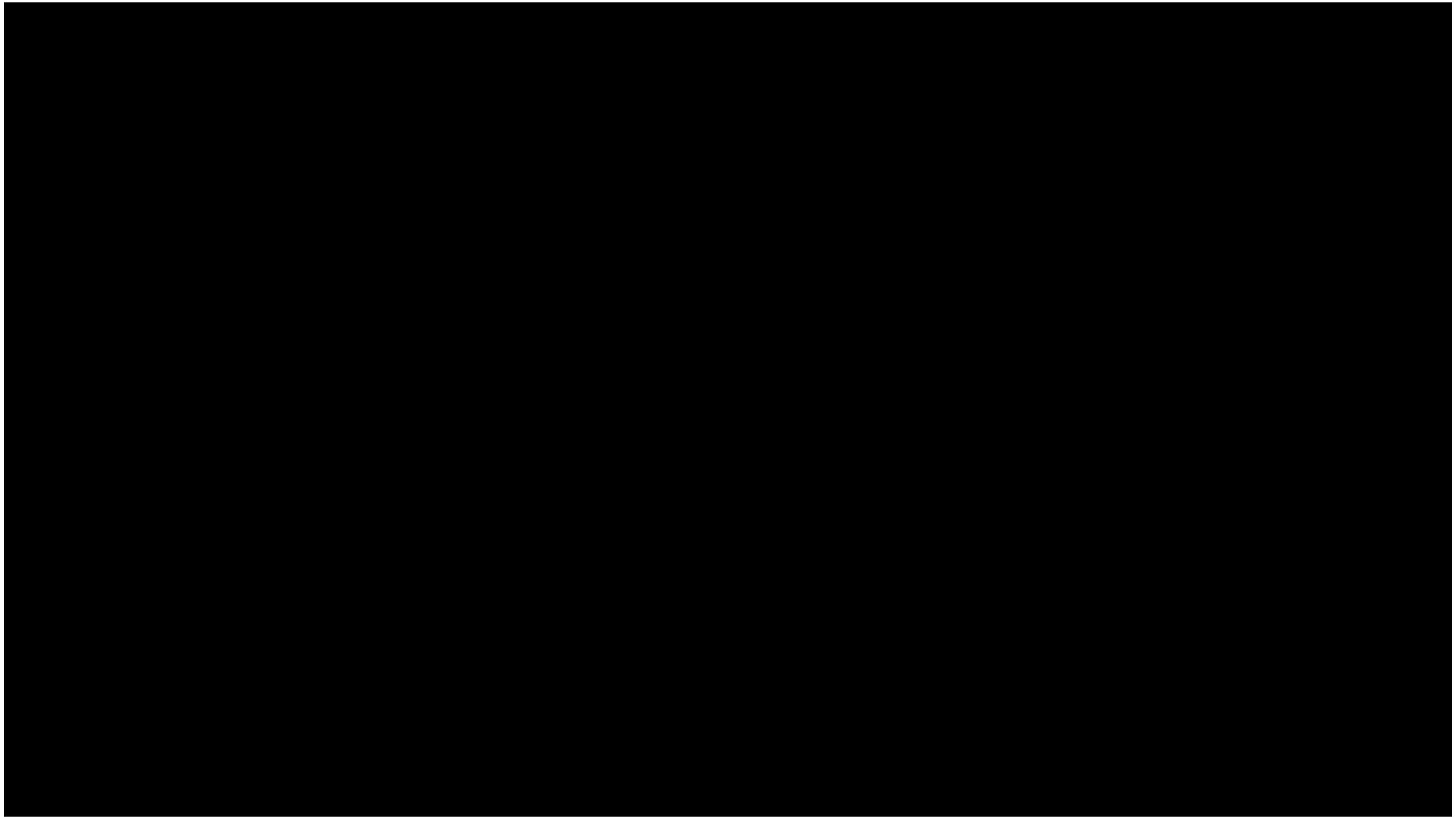




- **Software/Firmware**
  – Standard TI development environment
  – Real time operating system (RTOS), sensor APIs,
  – Communication services
- **Reference Applications (all real-time)**
  – Human activity recognition, gesture recognition, gait analysis

- **Energy-Harvesting & Management**
  – Photovoltaic (PV)-cells for ambient light
  – Piezo-electric materials for human motion
  – Optimum run-time enegy-allocation

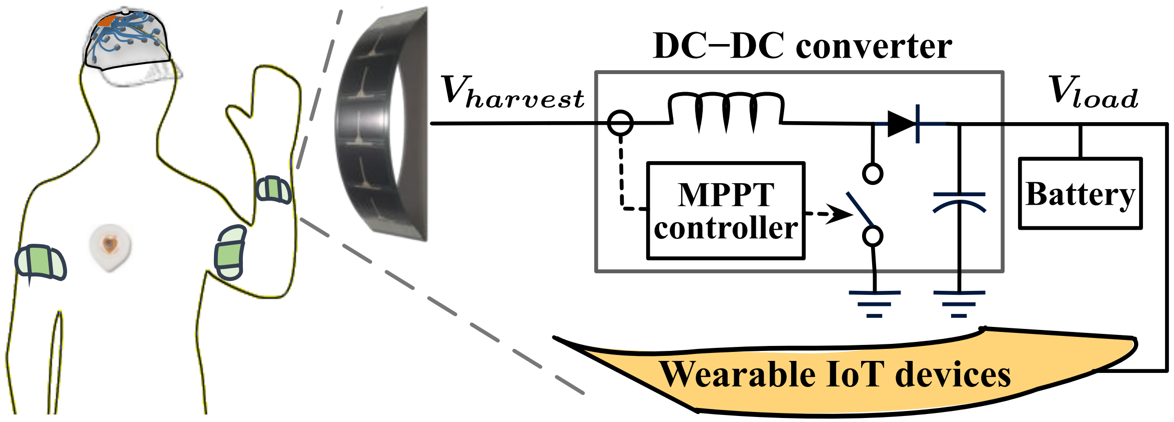
Example: Motion Energy Harvesting



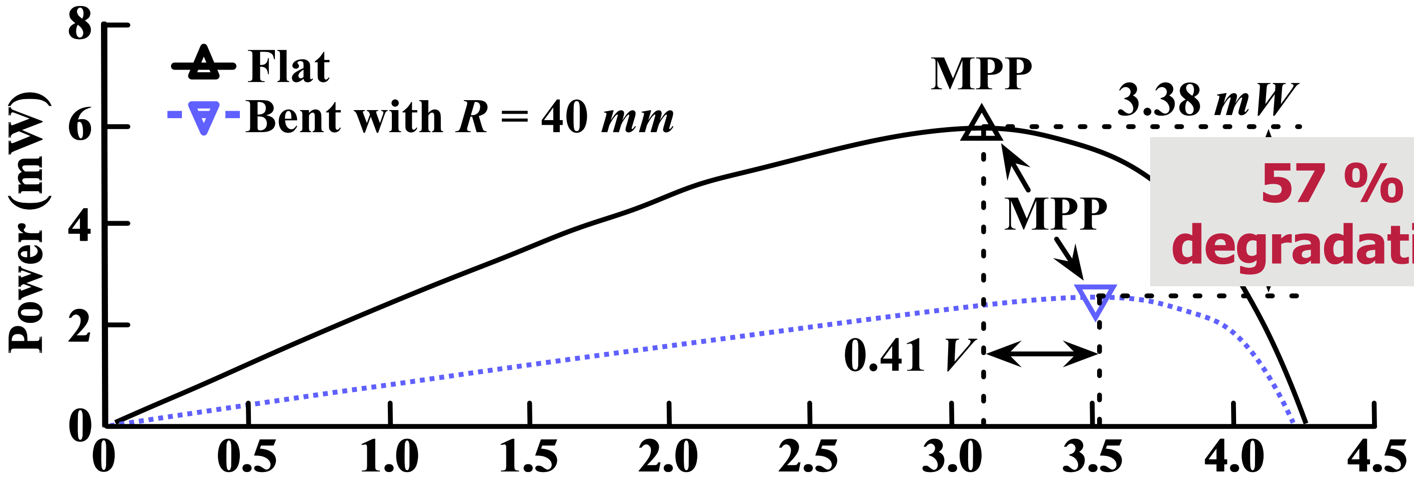
Y. Tuncel, et al., Towards wearable piezoelectric energy harvesting: modeling and experimental validation. In Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED), 2020

Energy Harvesting with Flexible Photo-Voltaic (PV) Cells

- **Energy harvesting for wearable devices**
  - Limited battery size and weight
- **PV-cell is one of the most widely used energy harvesting source**
  - Superior outdoor (10−100mW/cm^2) and good indoor (100 $\mu W/cm^2$) power density
  - Flexibility advantage for wearable applications



- **Output power determined by maximum power point (MPP)**
  - Bending has a significant impact on the harvested power



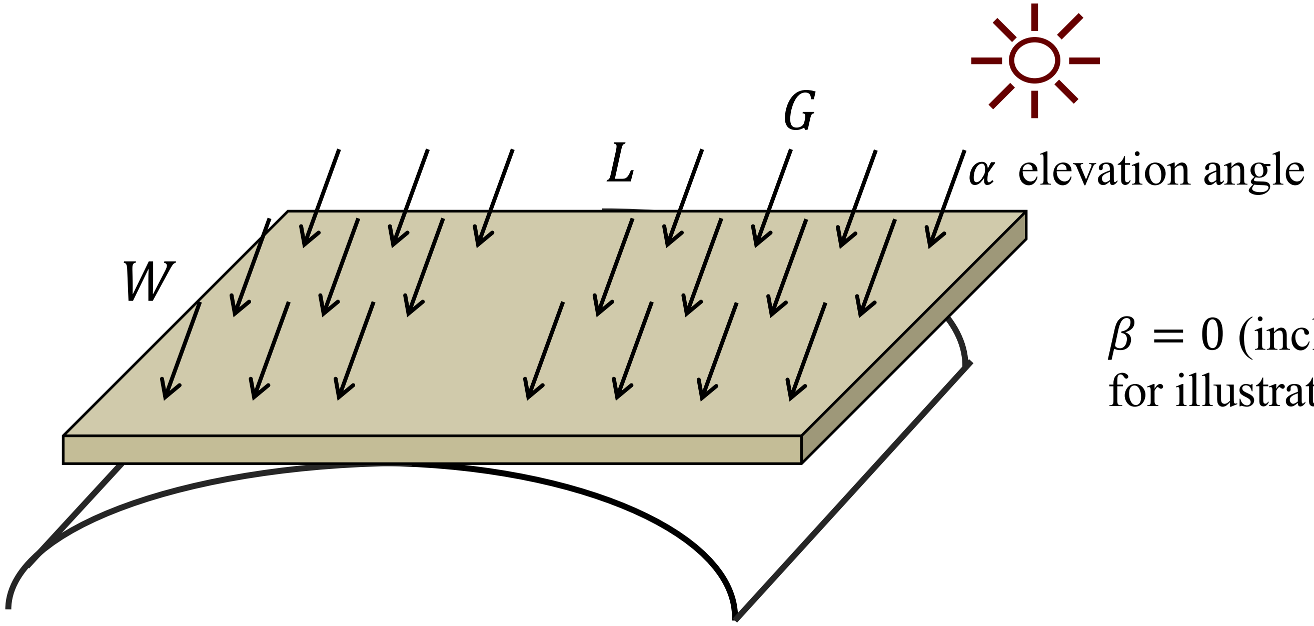
57 % degradation!

No analytical models that quantify the changes

Analytical Model with Bending

- **The amount of radiation received by the flat PV-cell**

$$\lambda_{flat} = L \cdot W \cdot G \cdot \sin(\alpha + \beta) \quad \Longleftarrow \quad \lambda = \int_{\frac{-L}{2}}^{\frac{L}{2}} \int_{0}^{W} G \cdot \sin(\alpha + \beta)\, dwdl$$



$\beta = 0$ (inclination angle)
for illustration

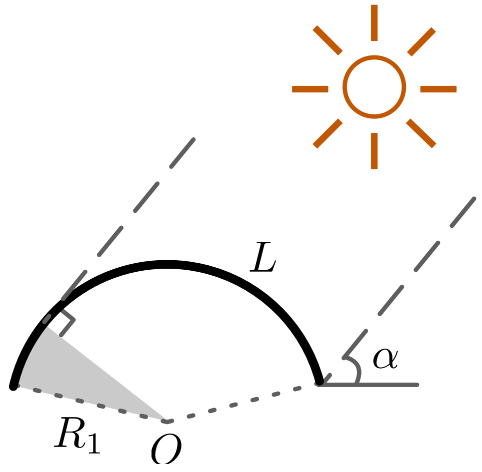
- **When a PV-cell is bent, the irradiation is not uniform across the bending axis**

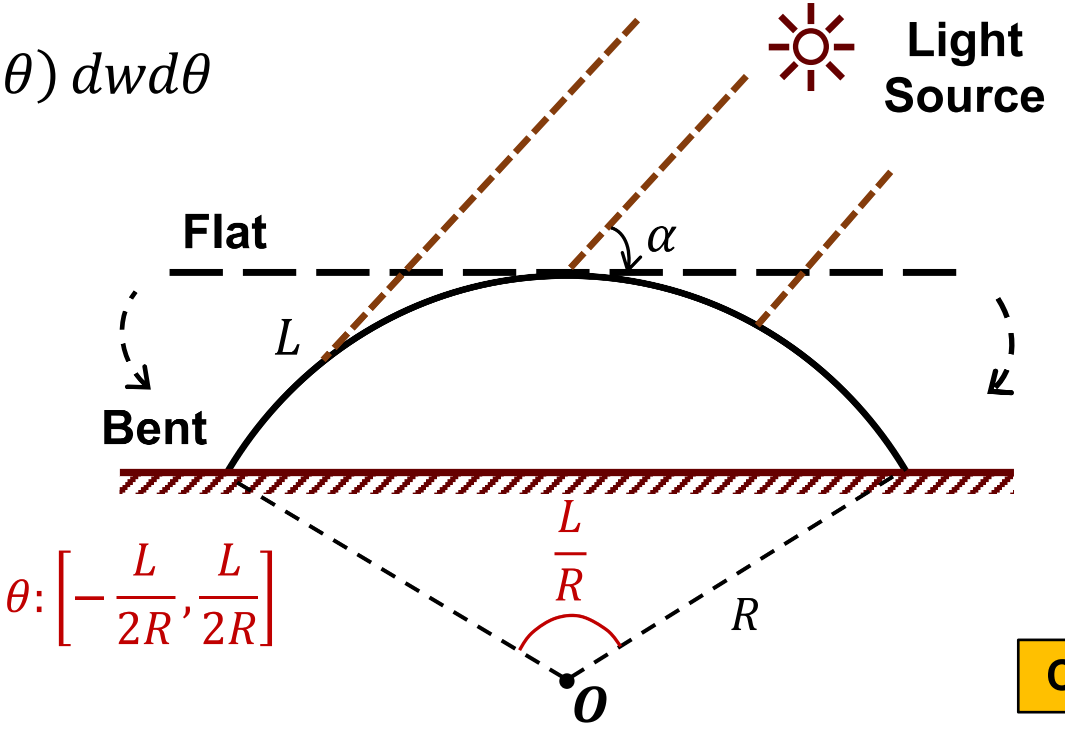
Analytical Model with Bending

- **When a PV-cell is bent with a radius of curvature R**
  - The length of an infinitesimal cross section of the arc $dl = R \cdot d\theta$
  - The limit of θ can be found as follows

$$-\frac{L}{2} = R \cdot \theta_{min} \Rightarrow \theta_{min} = -\frac{L}{2R}, \qquad \frac{L}{2} = R \cdot \theta_{max} \Rightarrow \theta_{max} = \frac{L}{2R}$$

$$\lambda_{bent} = \int_{\frac{-L}{2R}}^{\frac{L}{2R}} R \int_{0}^{W} G \cdot \sin(\alpha + \beta + \theta)\, dw d\theta$$





Core model

Irradiation Models under Shading Scenarios

1: No shading

$$\lambda_{bent.1} = \int_{\frac{-L}{2R_2}}^{\frac{L}{2R_2}} R_2 \int_0^W G \cdot \sin(\alpha+\beta+\theta)\, dw d\theta$$
$$= 2W \cdot G \cdot R_2 \sin(\alpha+\beta) \sin(\frac{L}{2R_2})$$

2: One-side shaded

$$\lambda_{bent.2} = \int_{-(\alpha+\beta)}^{\frac{L}{2R_1}} R_1 \int_0^W G \cdot \sin(\alpha+\beta+\theta)\, dw d\theta$$
$$= W \cdot G \cdot R_1 \left(1 - \cos\left(\alpha+\beta+\frac{L}{2R_1}\right)\right)$$

3: Both sides shaded

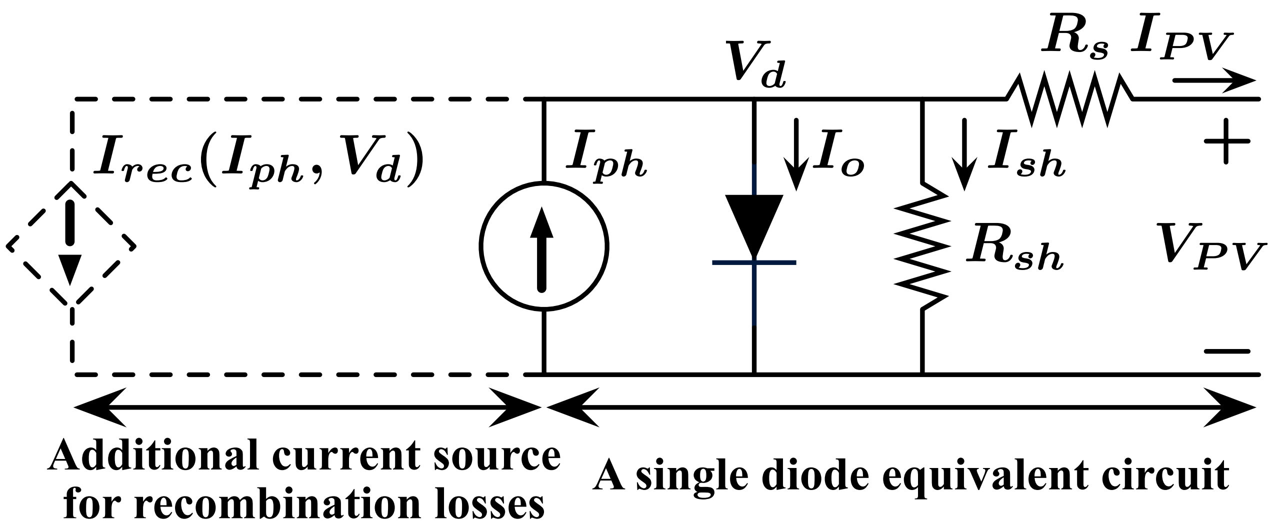
$$\lambda_{bent.3} = \int_{-(\alpha+\beta)}^{\pi-(\alpha+\beta)} R_3 \int_0^W G \cdot \sin(\alpha+\beta+\theta)\, dw d\theta$$
$$= 2W \cdot G \cdot R_3$$

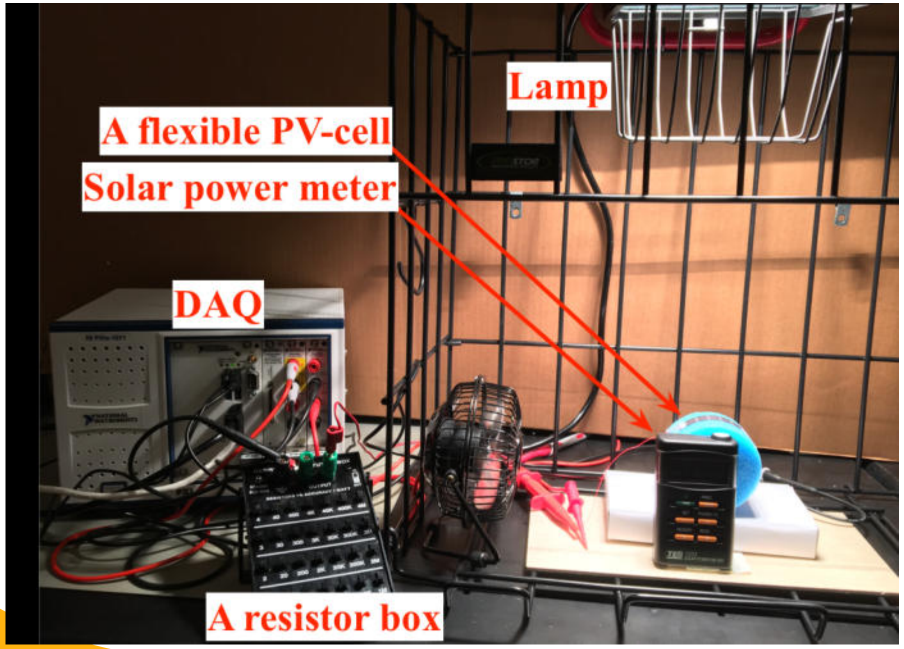
Single cell

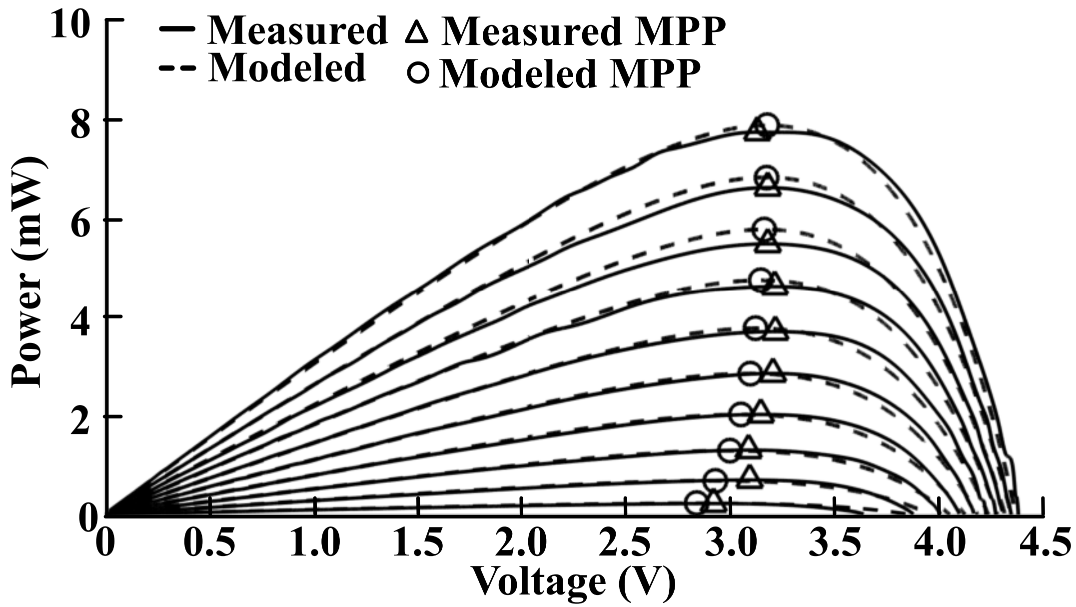
Core model

Current-Voltage Modeling and Validation

- **Equivalent circuit model for a <u>PV-cell</u>**
  - Need to consider recombination losses
  - Circuit parameters are functions of our irradiation model
- **Need to validate empirically**
  - Commercial *FlexSolarCells SP3-12*
  - Radiation intensity: 100~1000 W/m^2



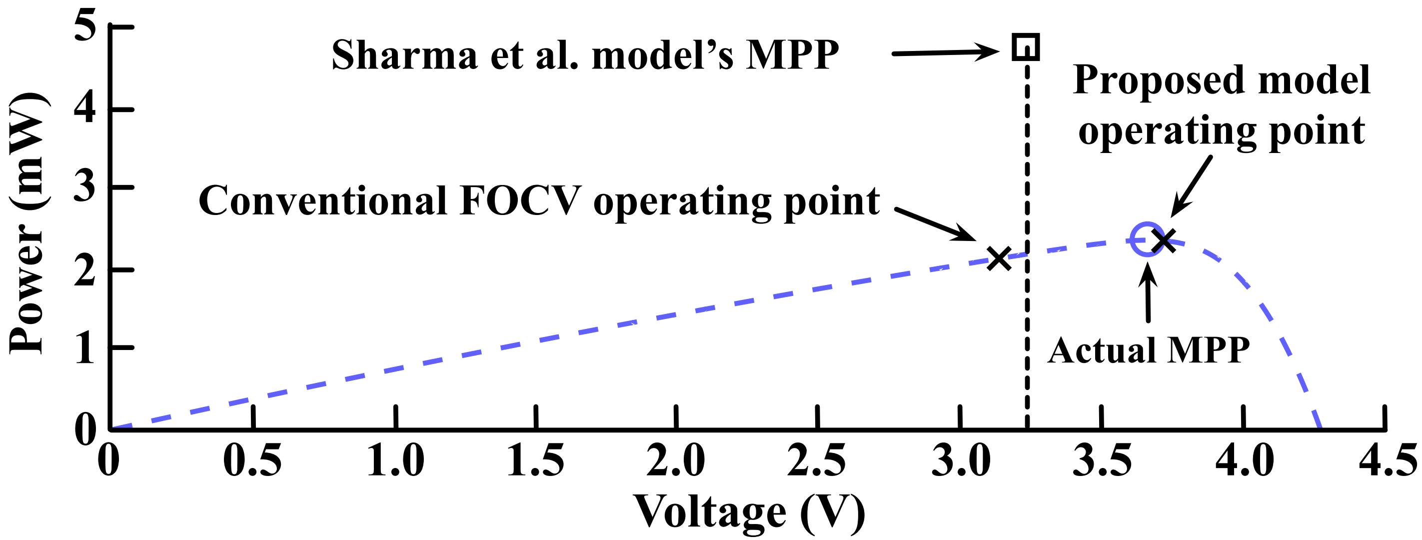




Relative percentage error of $P_{MPP} < 5.3\%$

Real-Life Impact: Maximum Power Point

- **Traditional fractional open-circuit voltage (FOCV) assumes V_{MPP}/V_{OC} does not vary significantly with radiation intensity**
- **Sharma et al. the only prior work that consider impact of bending (only experimentally)**



$G = 1000\ W/m^2, R = 40\ mm$

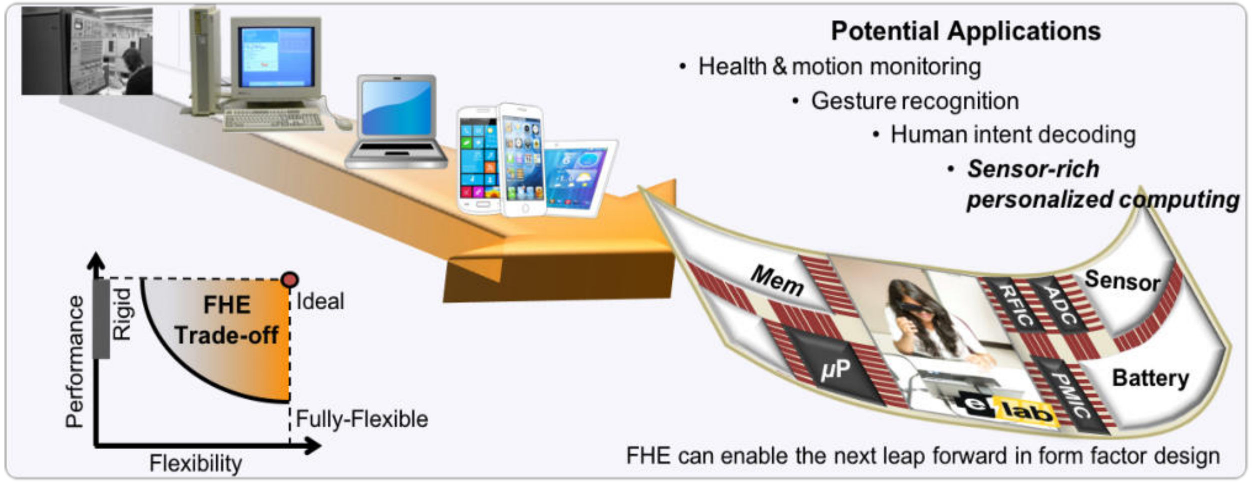
Proposed approact improves the MPPT gain by 19%

J. Park, et al. "Flexible PV-cell modeling for energy harvesting in wearable IoT applications." *ACM Transactions on Embedded Computing Systems (TECS)* 16.5s (2017): 1-20 **(CODES-ISSS Best Paper Award)**

P. Sharma, et al., 2014. A novel approach for maximum power tracking from curved thin-film solar photovoltaic arrays under changing environmental conditions. IEEE Transactions on Industry Applications 50, 6 (2014), 4142--4151

Summary & Conclusions

- **We are at the edge of next transition**
- **Self-powered mobile and wearable systems can enable the next leap forward**
- **Exciting opportunities in**
  - System (hardware/software) design
  - Processing / learning at the edge
  - Collaborative inference with the cloud



- **Our work encompasses**
  - Mobile and wearable computing systems
  - Energy harvesting and optimal management
  - SoC architecture design
  - Wearable sensor applications

https://sites.google.com/view/openhealth-wearable-health/home

Questions?

Thank You!