Minimizing Energy Consumptions of MPSoCs under Lifetime Reliability Constraint

Qiang XU

CUhk REliable computing laboratory (CURE)

Dept. of Computer Science & Engineering

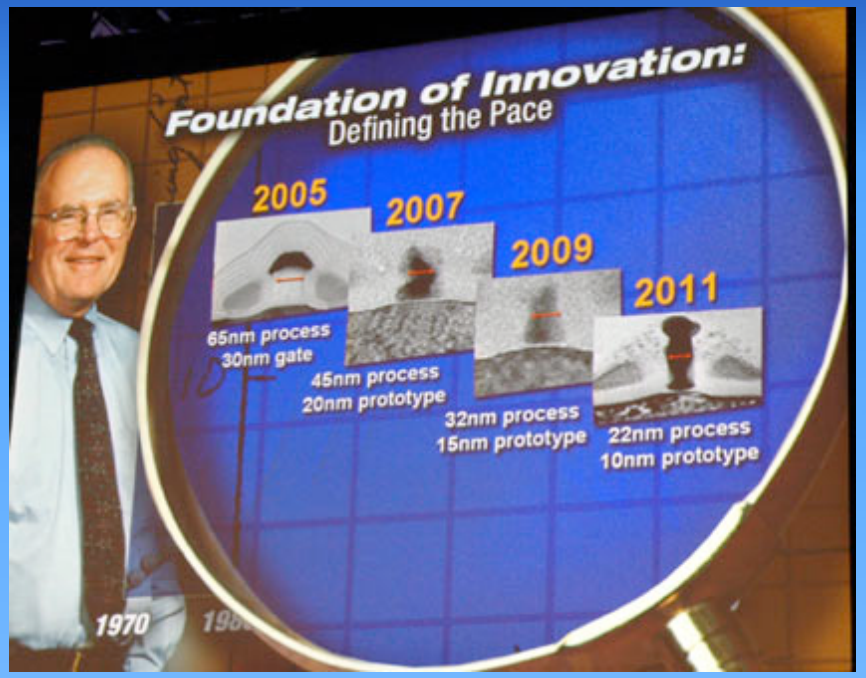
The Chinese University of Hong Kong

CUhk REliable Computing Laboratory

Agenda

- Introduction and motivation
- Problem formulation
- Analytical models
- Proposed task allocation and scheduling algorithms
- Experimental results
- Future work: ***product personalization?***
- Conclusion

Technology Scaling Continues ...



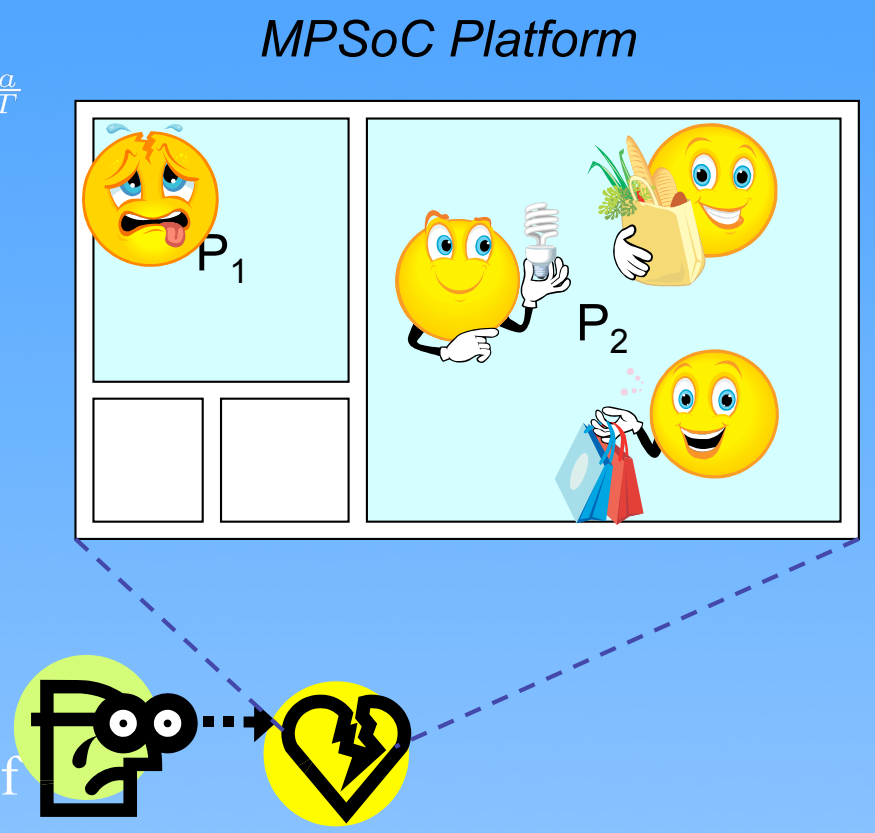
- **Minimizing energy consumption is a critical objective for embedded system designs**
- **Lifetime reliability threats have become serious concerns**
  - ✓ Electromigration (EM) on the interconnects
  - ✓ Time dependent dielectric breakdown (TDDB) in the gate oxides
  - ✓ Negative bias temperature instability (NBTI) on PMOS transistors

Why Explicitly Considering Lifetime Reliability Issue in Embedded System Design?

- **Electromigration**

$$MTTF_{EM} \propto (V_{dd} \times f \times p_i)^{-2} e^{\frac{E_a}{kT}}$$

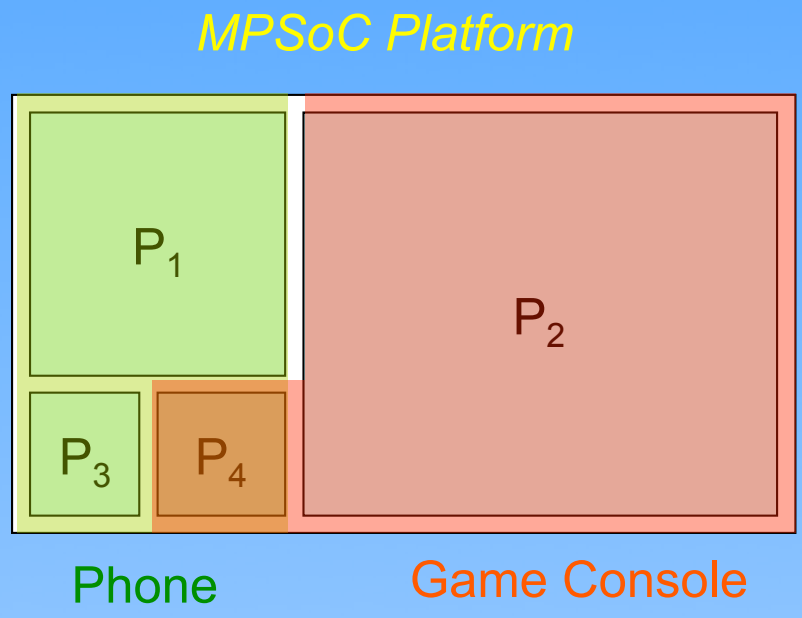
- **Suppose**
  - ✓ $f_1 = 2f_2$
  - ✓ All other parameters are the same
- **A recent work in DATE'09**
  - ✓ Efficient and effective MTTF estimation
  - ✓ Try to maximize the service life of single-mode systems
  - ✓ Does not consider energy issues



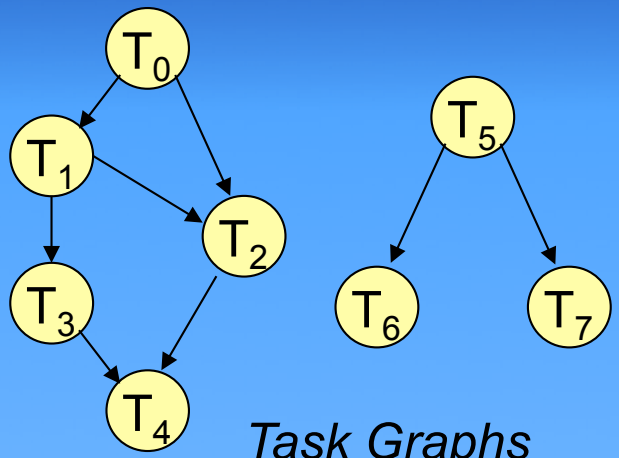
Motivation

- **Existing design targets at reducing overall energy consumption**
- **MPSoC service life, however, depends on the component with the minimum lifetime reliability**

- **Modern smart phone**
  - ✓ Communication service
  - ✓ MP3 player
  - ✓ Video decoder
  - ✓ Game console
  - ✓ Digital camera
  - ✓ GPS navigator
  - ✓ …

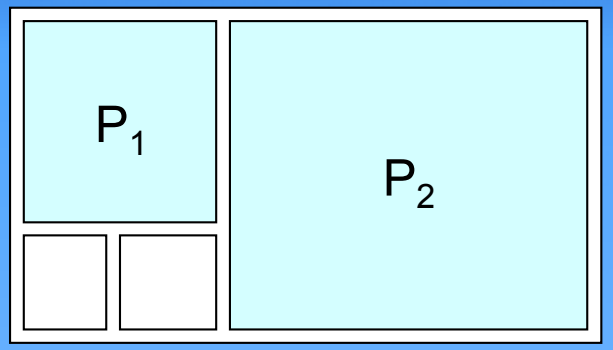


Problem Formulation



Task Graphs

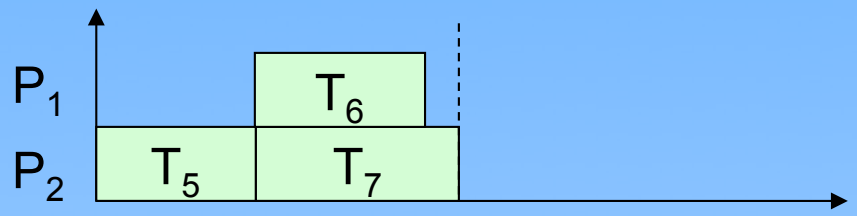
Binding & Scheduling



MPSoC Platform

- **Output**

Objective: To minimize energy consumption for multi-mode MPSoCs under performance and lifetime reliability constraint.



Periodical Schedules

Performance and Energy Model for DVS-Enabled Processors

- **Power dissipation**

$$P = P_{on} + C_{eff} \cdot V_{dd}^2 \cdot f + (V_{dd} \cdot K_1 e^{K_2 V_{dd}} + |V_{bs}| \cdot I_j)$$

dynamic power consumption

leakage power consumption

- **Voltage scaling parameter** $\rho \in [0, 1]$
- **Scaled operating frequency** $f(\rho) = \frac{(\rho V_{dd} - V_{th})^\kappa}{L_d K_3}$
- **Scaled power consumption**

$$P(\rho) = P_{on} + C_{eff} \cdot \rho^2 V_{dd}^2 \cdot f(\rho) + \rho V_{dd} \cdot K_1 e^{K_2 \cdot \rho V_{dd}} + |V_{bs}| \cdot I_j$$

- **Scaled execution time**

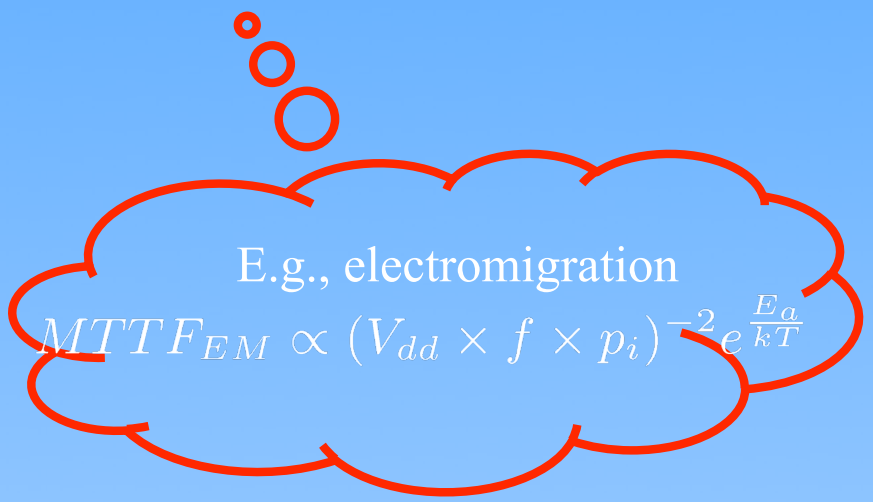
$$w(\rho) = \frac{(V_{dd} - V_{th})^\kappa}{(\rho V_{dd} - V_{th})^\kappa} \cdot w$$

Lifetime Reliability Model

- $A_{i,k}$ – **aging effect of processor** k **in an operational cycle**
- **System reliability at the end of target service life** L

$$R_i = e^{-\sum_k (\frac{A_{i,k}}{\max_j \{d_{i,j}\}} \cdot L)^{\beta_k}}$$

- $A_{i,k}$ **is approximated with its scaled execution time and steady temperature, substituting into existing failure models**



Proposed Algorithm for Single-Mode System: Simulated Annealing

- **Solution representation**
  - ✓ The most straightforward one
    - (*schedule order sequence*; *resource binding sequence*)
    - Huge design exploration space $v_i! \cdot \ell^{v_i}$
  - ✓ Energy consumption and lifetime reliability are almost independent of schedule orders
    - Use *resource binding sequence* only as representation
    - Design exploration space shrinks to ℓ^{v_i}
- **Cost function**

$$C_i = \gamma \cdot 1_{\{\exists j : m_{i,j} > d_{i,j}\}} + (1 - R_i) \times E_i$$

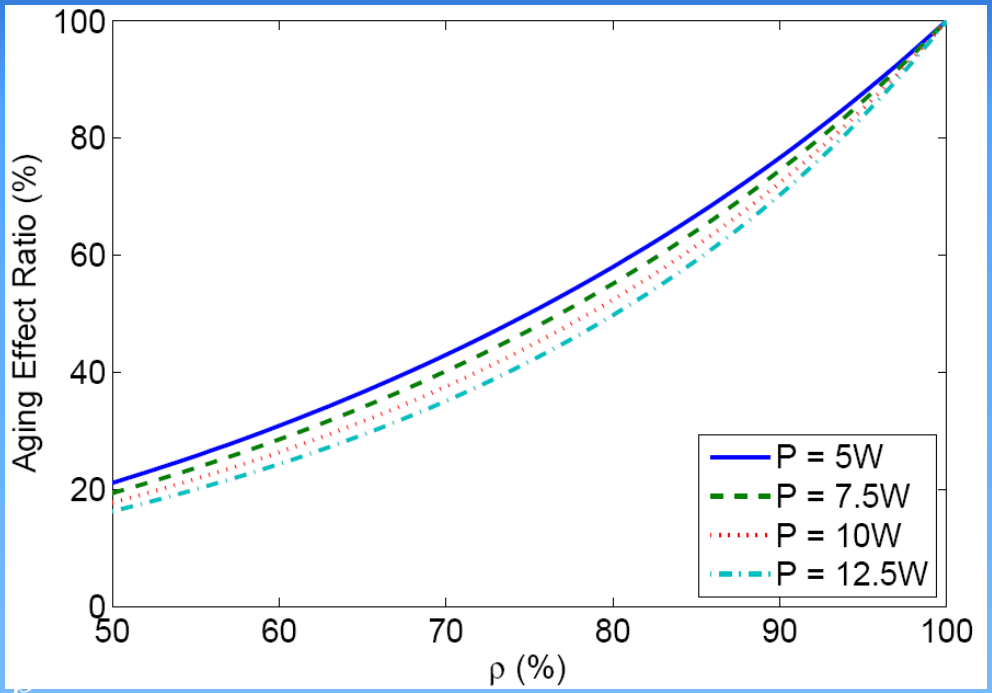
Significant large number

Some performance constraints are violated

Lifetime reliability

Energy consumption

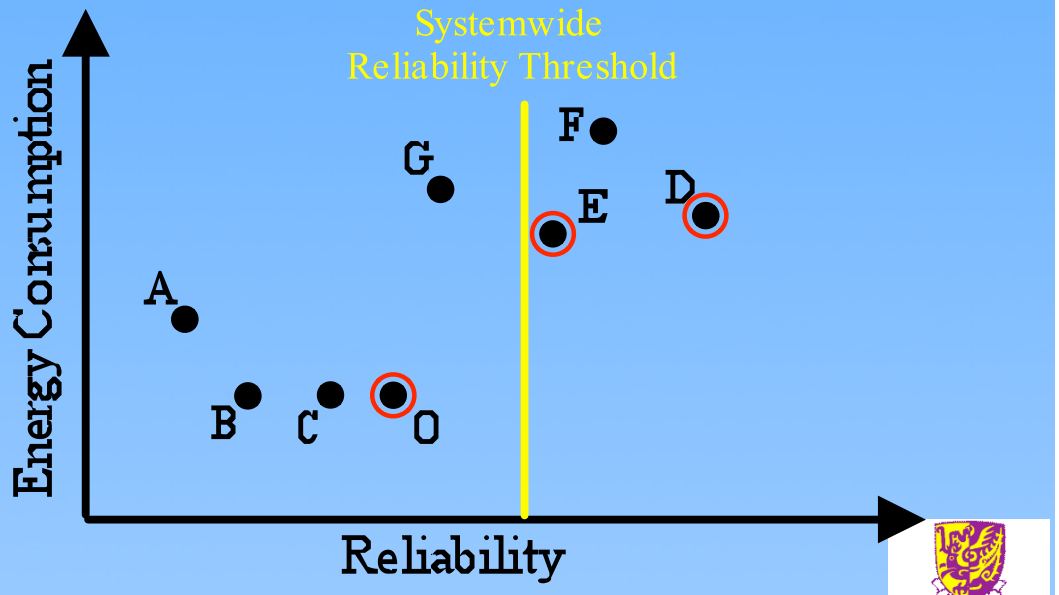
Proposed Algorithm for Single-Mode System: Simulated Annealing



- Compute the possible task execution time range and associated voltage scaling parameter
- Choose the voltage/frequency with best energy reduction and reliability enhancement

Algorithm for Multi-Mode Embedded Systems: Feasible Solution Set

- **Set X – all task schedules for a certain task graph**
- **Subset Y satisfies**
  - ✓ **Internal stability** Given two solutions $u, v \in Y$, if u consumes more energy than v, it must result in higher lifetime reliability at the target service life, and vice versa
  - ✓ **External stability** For any solution $w \in X \backslash Y$, there exists at least one solution $u \in Y$ such that u consumes less energy and have higher lifetime reliability than w
- **We refer to Y as the feasible solution set, denoted by $\mathcal{F}$**



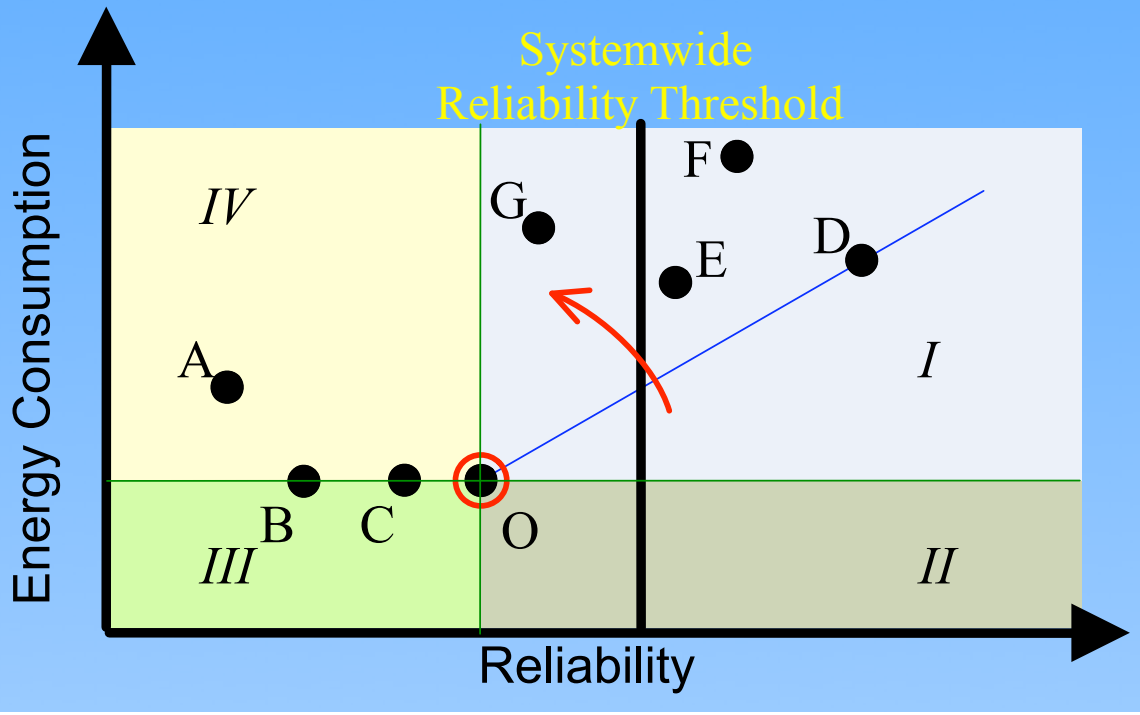
Algorithm for Multi-Mode Embedded Systems: Searching Procedure

1 Build an initial solution **X** and initialize $\mathcal{P}$ as {**X**}
2 **While** $T > T_{end}$
3 **For** the iteration i from 1 to I
4 **X'** ← Random_Move(**X**)
5 C_{old} ← Cal_Cost(**X**)
6 C_{new} ← Cal_Cost(**X'**)
7 **If** $C_{new} < C_{old}$ **or** $\exp\left(\frac{C_{old} - C_{new}}{T}\right) > rand()$
8 **X** ← **X'** // accept **X'**
9 **If** **X'** $\notin \mathcal{P}$ **and** **X'** meet performance constraint
10 **If** static identification
11 $\mathcal{P}$ ← Include($\mathcal{P}$, **X'**)
12 **Else If** dynamic identification
13 $\mathcal{P}$ ← Identify($\mathcal{P}$, **X'**)
14 $T \leftarrow T \times R_{cooling}$

15 **If** static identification
16 $\mathcal{F}$ ← Identify($\mathcal{P}$)
17 **Else If** dynamic identification
18 $\mathcal{F}$ ← $\mathcal{P}$

Algorithm for Multi-Mode Embedded Systems: Feasible Solution Set Identification

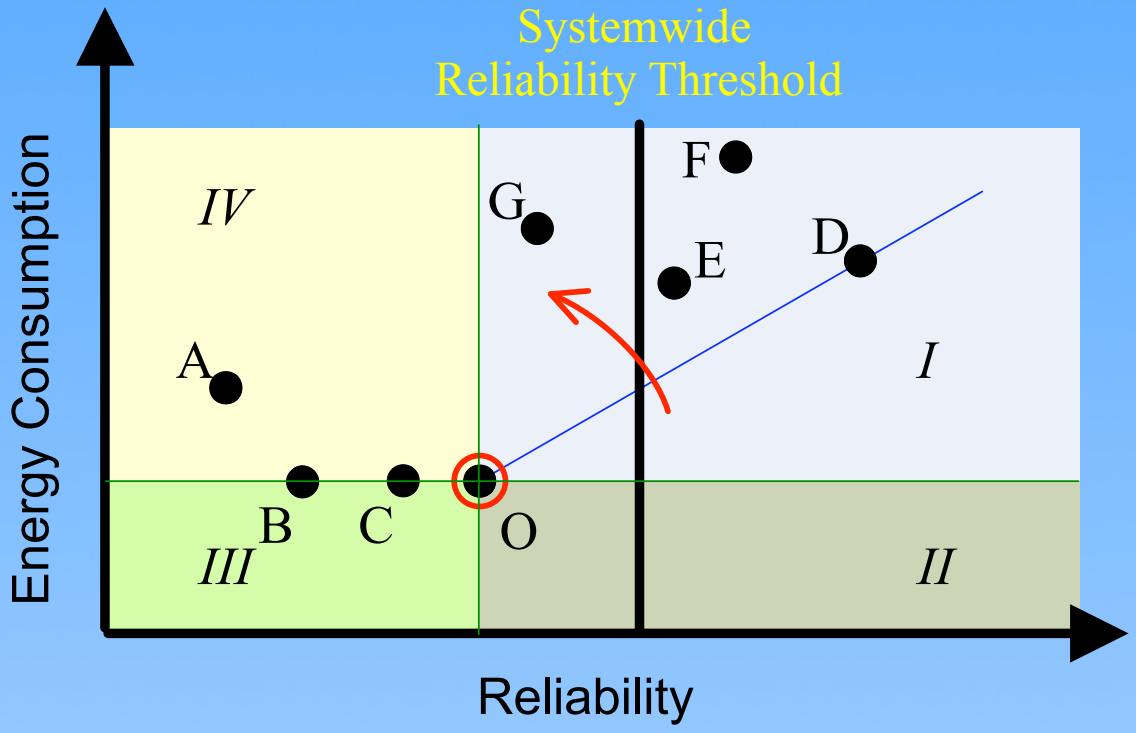
- **Static strategy – all solutions have been found out beforehand**
  - ✓ Original point **O**
    - The point with the lowest energy consumption and, in case of ties, with the highest lifetime reliability
    - Must be a feasible solution
  - ✓ With respect to a point, dividing the plane into four domains *I*, *II*, *III*, *IV*
  - ✓ Sweep domain *I* with respect to point **O** in a counterclockwise manner



CUhk REliable Computing Laboratory

Algorithm for Multi-Mode Embedded Systems: Feasible Solution Set Identification

- **Static strategy – all solutions have been found out beforehand**
  - ✓ Sweep domain *I* with respect to point **O** in a counterclockwise manner
  - ✓ A new solution **N** is a feasible one if and only if it is in domain *I* or *III* of all elements in $\mathcal{F}$



original $\mathcal{F}$	new solution	updated $\mathcal{F}$
{O}	D	{O,D}
{O,D}	E	{O,D,E}
{O,D,E}	F	{O,D,E}
{O,D,E}	G	{O,D,E}

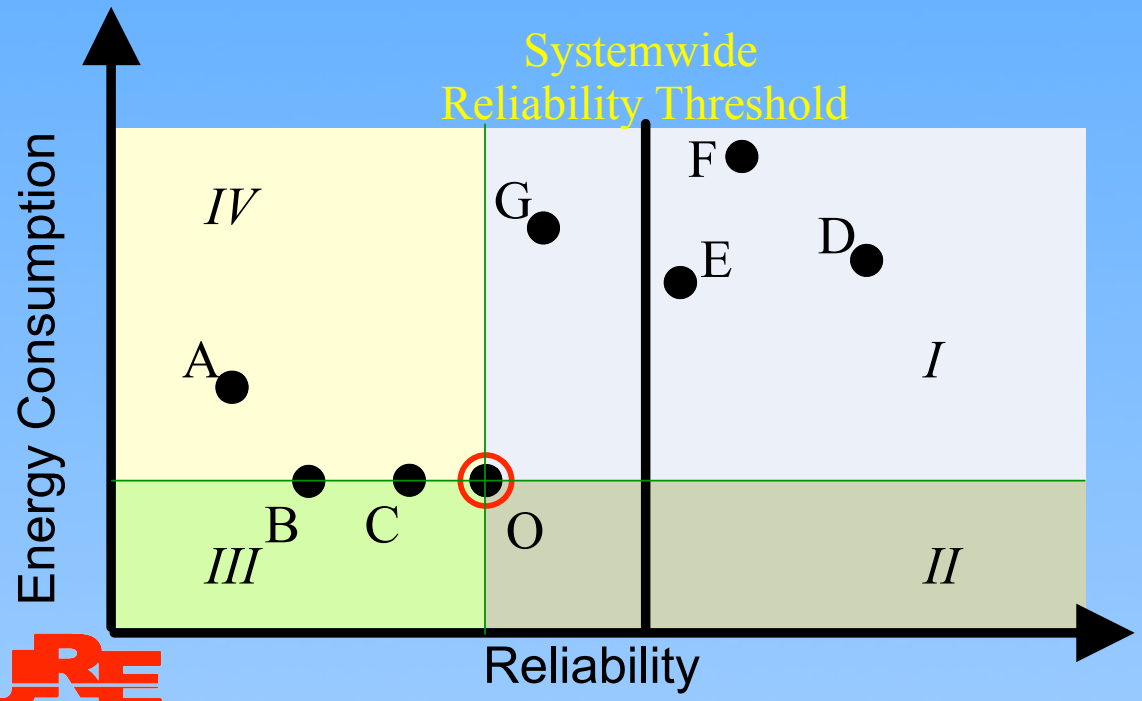
CUhk REliable Computing Laboratory

Algorithm for Multi-Mode Embedded Systems: Feasible Solution Set Identification

- **Static strategy – all solutions need to be found beforehand**
  - ✓ Heavy memory overhead
- **Dynamic strategy – identify solutions in a random order**
  - ✓ Every newfound solution is processed according to
    - **Rule 1** If the new solution is in domain *I* or *III* of ALL elements in set $\mathcal{F}$, it should be included into the feasible set $\mathcal{F}$
    - **Rule 2** If the new solution is in domain *II* of ANY solution **X** (where $\mathbf{X} \in \mathcal{F}$), we include the new solution into $\mathcal{F}$ and at the same time eliminate **X** from $\mathcal{F}$
    - **Rule 3** If the new solution is in domain *IV* of ANY solution **X** (where $\mathbf{X} \in \mathcal{F}$), we ignore the new solution

Algorithm for Multi-Mode Embedded Systems: Feasible Solution Set Identification

- **Dynamic strategy – identify solutions in a random order**
  - ✓ **Rule 1** If the new solution is in domain *I* or *III* of ALL elements in set $\mathcal{F}$, it should be included into the feasible set $\mathcal{F}$
  - ✓ **Rule 2** If the new solution is in domain *II* of ANY solution **X** (where $\mathbf{X} \in \mathcal{F}$), we include the new solution into $\mathcal{F}$ and at the same time eliminate **X** from $\mathcal{F}$
  - ✓ **Rule 3** If the new solution is in domain *IV* of ANY solution **X** (where $\mathbf{X} \in \mathcal{F}$), we ignore the new solution



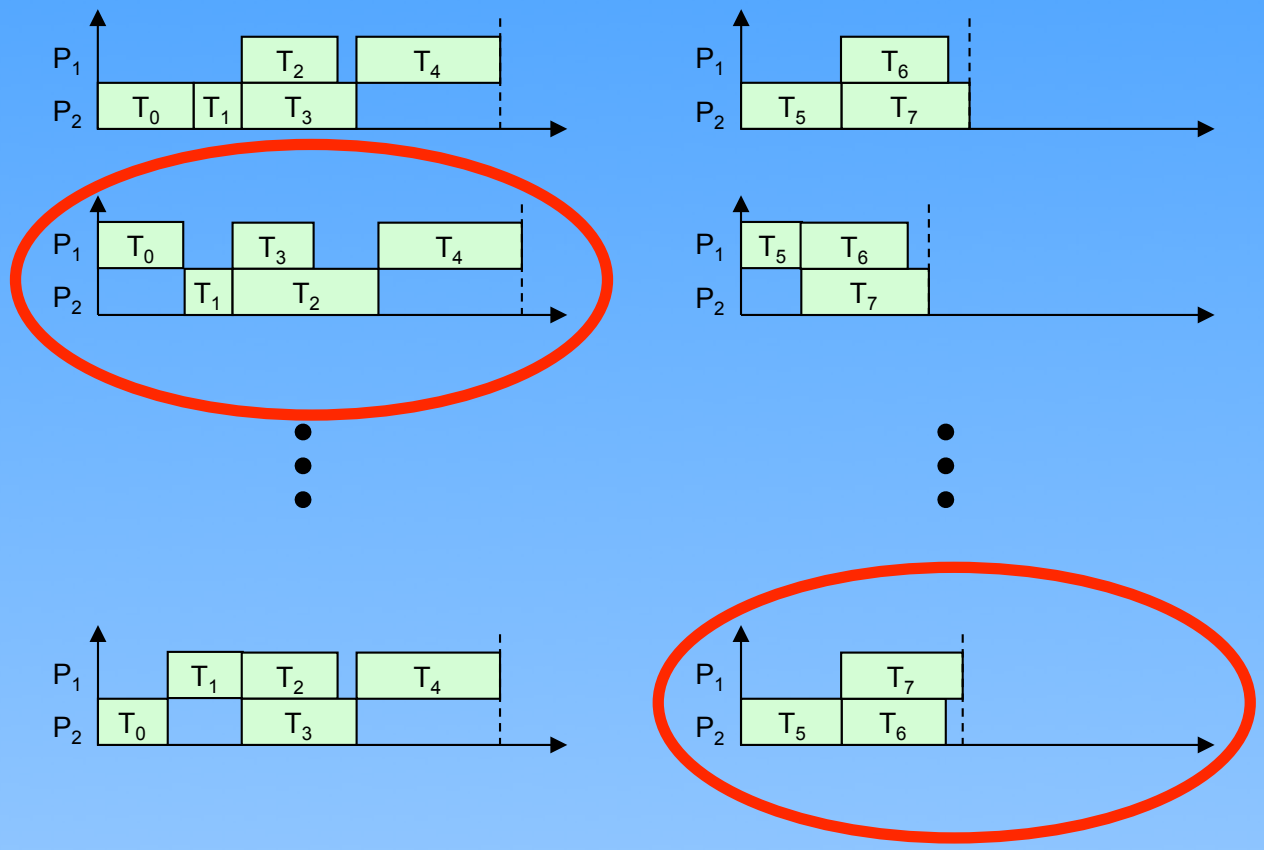
original $\mathcal{F}$	new solution	updated $\mathcal{F}$
Ø	A	{A}
{A}	E	{A,E}
{A,E}	F	{A,E,F}
{A,E,F}	G	{A,E,F}
{A,E,F}	C	{C,E,F}
{C,E,F}	O	{O,E,F}
{O,E,F}	D	{O,E,D}
{O,E,D}		

CUhk REliable Computing Laboratory

Algorithm for Multi-Mode Embedded Systems: Multi-Mode Combination

Execution mode 1

Execution mode 2



$$
\begin{aligned}
\min \quad & \mathbf{E}[E(Y_1, Y_2, \cdots, Y_n)] \\
\text{st.} \quad & \mathbf{E}[R(Y_1, Y_2, \cdots, Y_n)] \geq \eta \\
& \forall i : \textstyle\sum_\ell 1_{\{\ell \Rightarrow i\}} = 1 \\
& \forall i, \ell : 1_{\{\ell \Rightarrow i\}} = 0 \text{ or } 1
\end{aligned}
$$

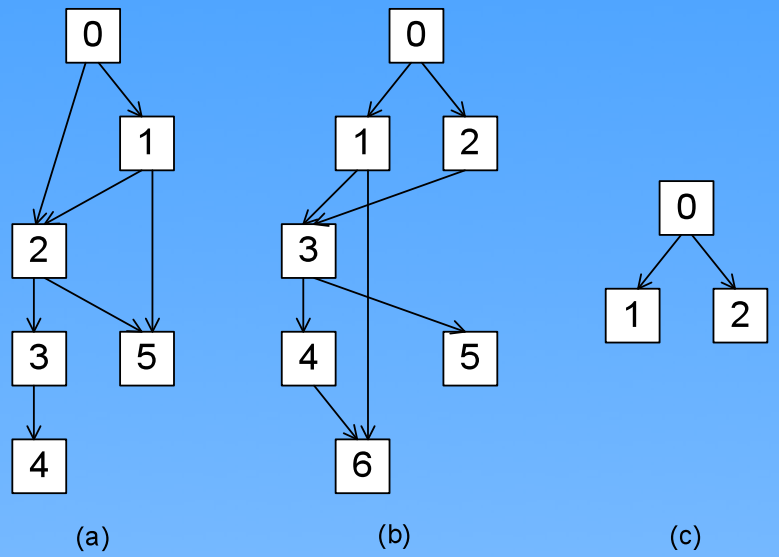
CUhk REliable Computing Laboratory

Experimental Setup

- **Task graphs**
  - ✓ Generated by TGFF
- **Hypothetical MPSoC platforms**
  - ✓ Heterogeneous and homogeneous
- **Power dissipation**
  - ✓ Randomly generated
  - ✓ The range is set according to state-of-the-art technology
- **Failure model**
  - ✓ Electromigration
  - ✓ Applicable for any failure model or the combination of multiple failures
- **Greedy heuristic**
  - ✓ Lack of prior work on the same topic
  - ✓ Step 1: list scheduling
    - Extend Highest Levels First with Estimated Times for heterogeneous systems
    - Shorten the schedule length
    - Reduce energy consumption
  - ✓ Step 2: iteratively reassign tasks to meet the reliability requirement
- **Proposed single-mode approach**
- **Proposed multi-mode approach**

Case Study

- **Task graphs**

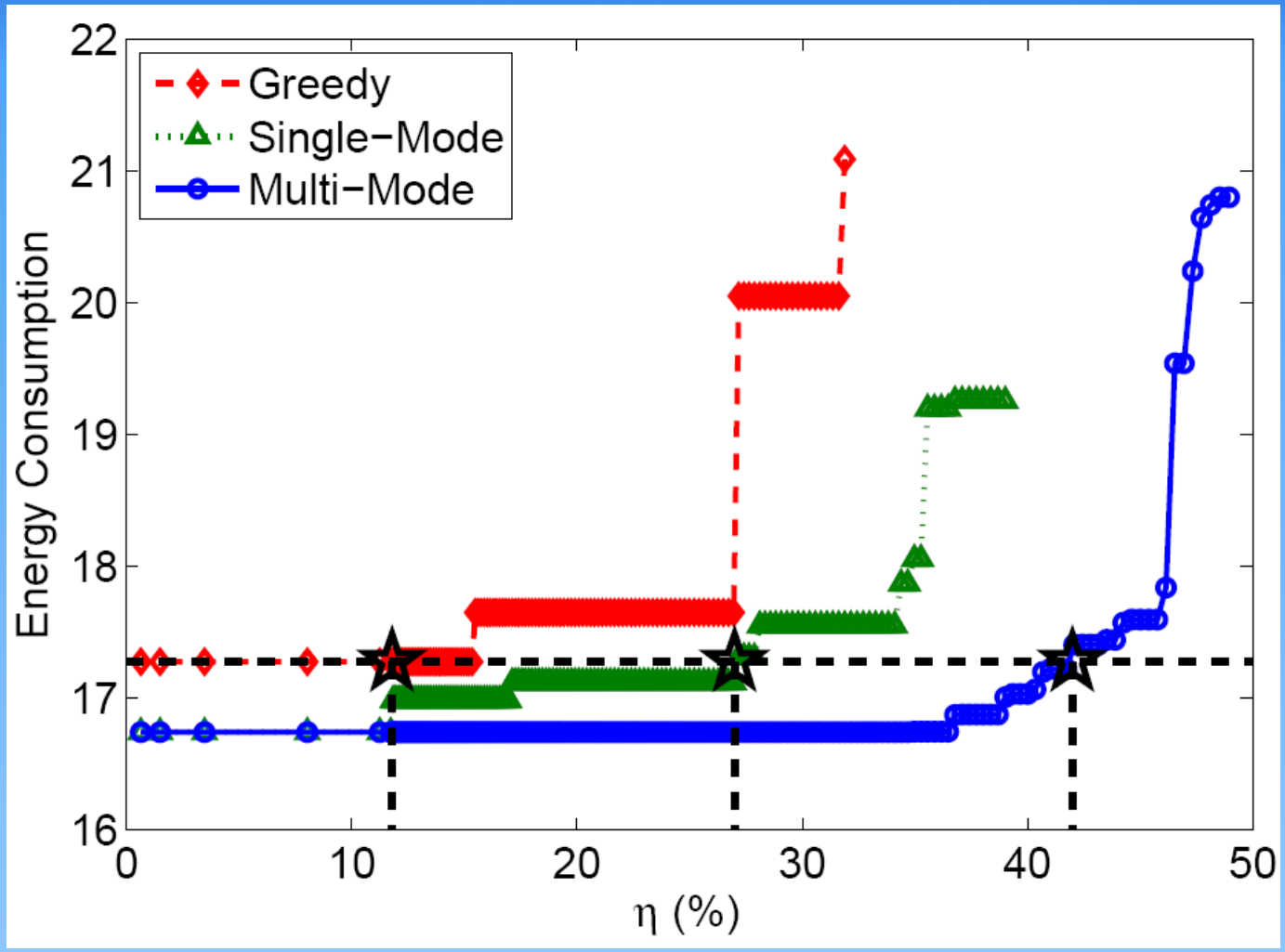


  - ✓ Probabilities: 0.3, 0.3, 0.4
- **2 processor cores**
  - ✓ $\beta = 1.5, 2$
- **Reliability requirement**
  - ✓ $R_i \geq 36.8\%$ (or e^{-1})

Task Graph Label	Sol. Index	$R_i(\%)$	E_i	Resource Binding Sequence
(a)	1	39.16	23.18	(0, 0, 1, 0, 1, 1)
	2	36.70	22.99	(1, 0, 1, 0, 1, 0)
	3	34.91	22.37	(0, 0, 1, 1, 1, 1)
	4	34.18	21.31	(1, 0, 1, 0, 1, 1)
	5	27.92	20.50	(1, 0, 1, 1, 1, 1)
	6	17.05	20.06	(1, 1, 1, 0, 1, 1)
	7	11.80	19.25	(1, 1, 1, 1, 1, 1)
(b)	1	65.09	15.44	(1, 0, 1, 1, 1, 0, 0)
	2	59.19	15.36	(1, 1, 1, 1, 0, 0, 0)
	3	56.94	14.92	(1, 0, 1, 1, 0, 0, 0)
	4	47.54	14.91	(1, 0, 0, 1, 0, 0, 0)
	5	45.49	14.49	(1, 0, 1, 0, 0, 0, 0)
	6	36.51	14.48	(1, 0, 0, 0, 0, 0, 0)
(c)	1	44.05	23.04	(1, 1, 0)
	2	41.98	22.56	(0, 1, 0)
	3	39.77	19.89	(1, 0, 1)
	4	35.33	17.03	(1, 0, 0)
	5	27.10	16.56	(0, 0, 0)

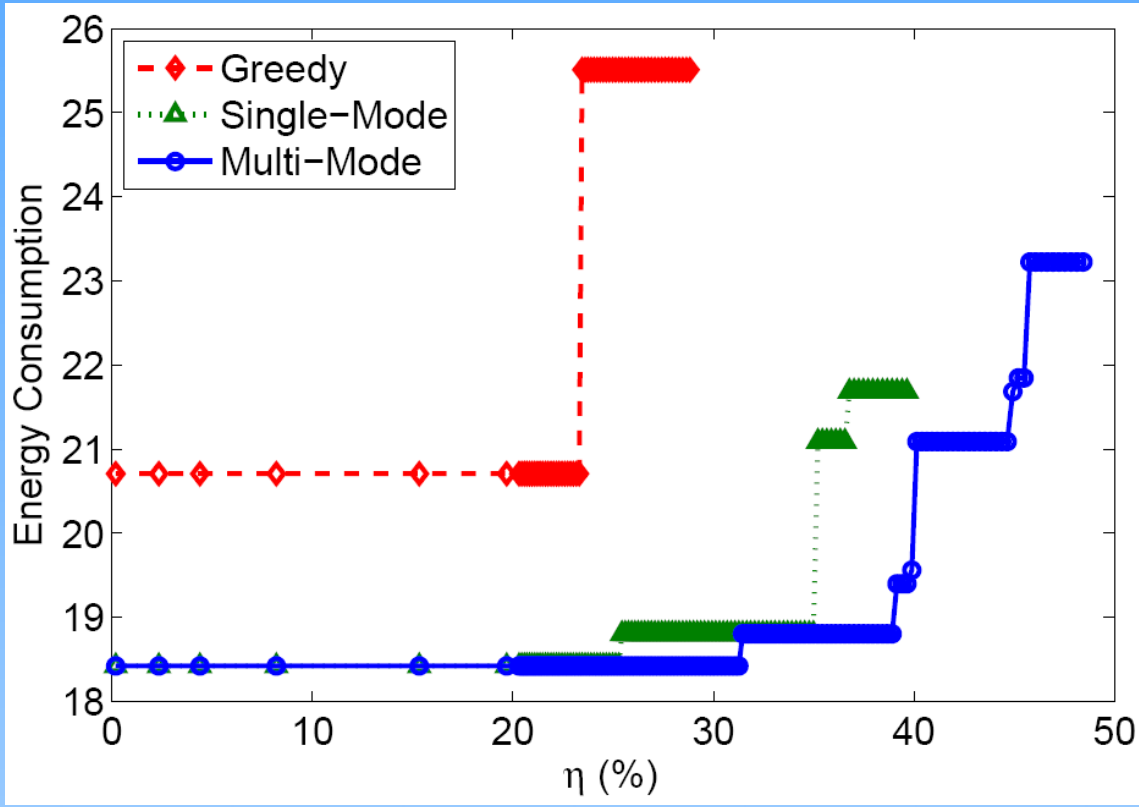
Greedy	Single Mode	Multi Mode
—	19.255	16.875

Sensitivity Analysis



Extensive Results

- **5 execution modes**
  - ✓ Task quantities: 69, 7, 14, 3, and 29
- **8-core heterogeneous MPSoC**
  - ✓ β = 2, 4, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5

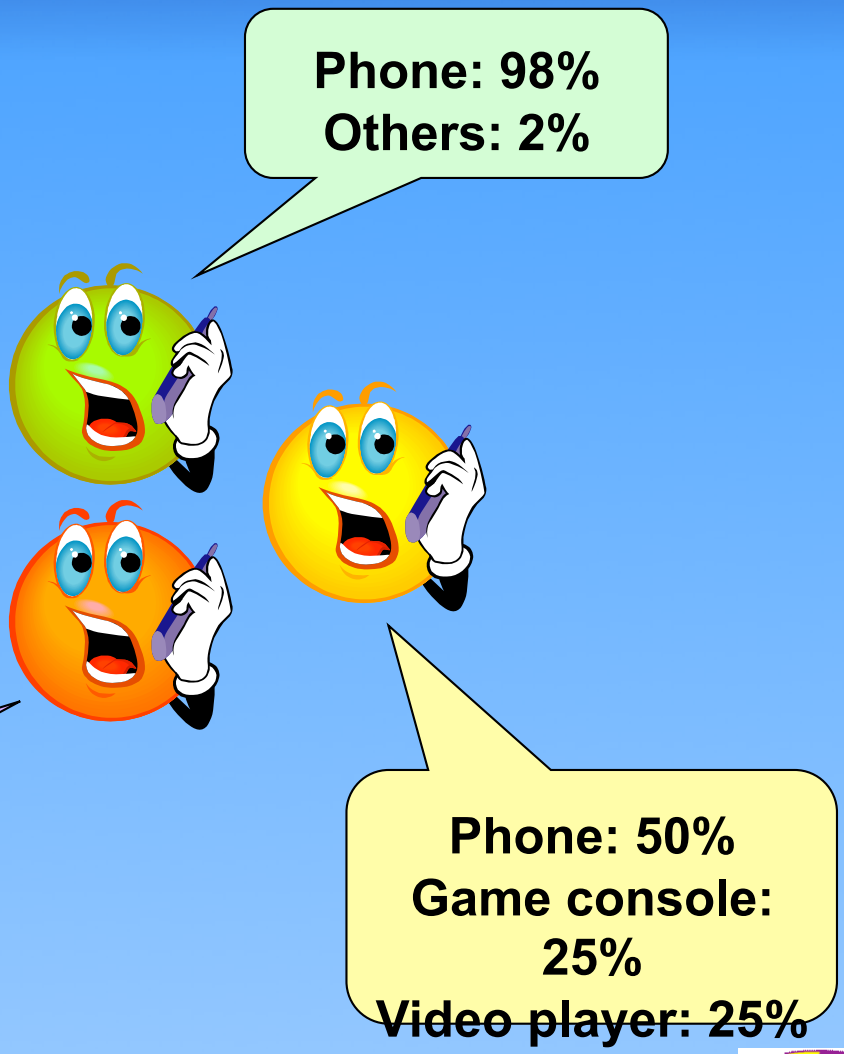


Extensive Results

- **5 execution modes**
  - ✓ Task quantities: 69, 7, 14, 3, and 29
- **8-core homogeneous MPSoC**
  - ✓ $\beta = 1.5$
- **Observations**
  - ✓ Three approaches result in same energy consumption when $\eta = 27.7\%$
  - ✓ Only the proposed multi-mode approach is able to achieve higher lifetime reliability, $\eta = 37.3\%$

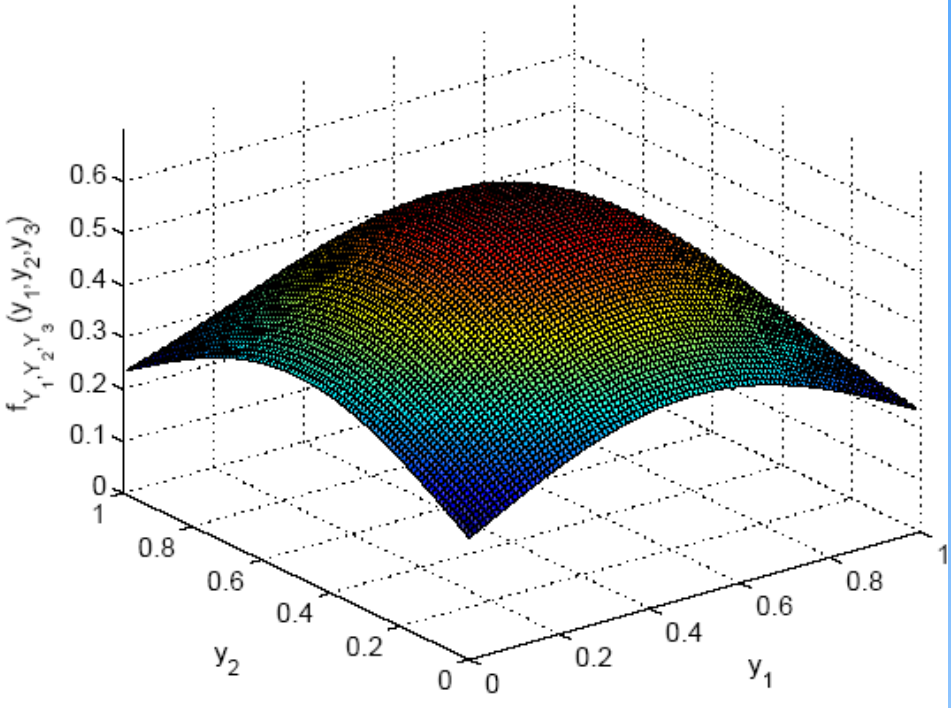
Product Personalization

- **Modern smart phone**
  - ✓ Communication service
  - ✓ MP3 player
  - ✓ Video decoder
  - ✓ Game console
  - ✓ Digital camera
  - ✓ GPS navigator
  - ✓ …

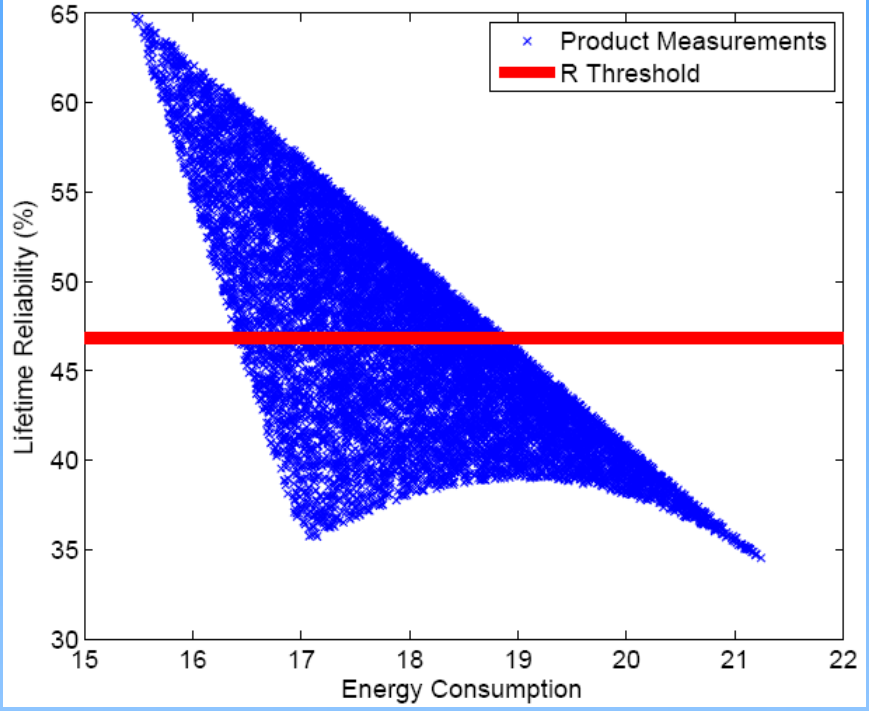


The Impact of Usage Strategies

- MPSoC with three operational modes
- Determine task schedules such that the expected energy consumption is minimized and expected lifetime reliability is no less than 46%



Joint Probability Density

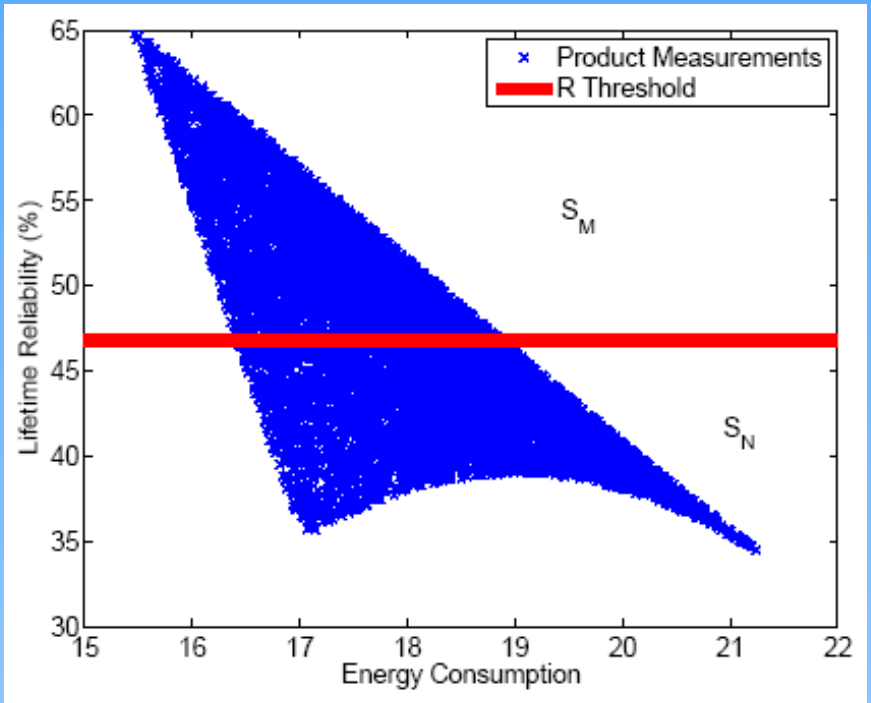


Product Measurement Deviation for 10,000 products

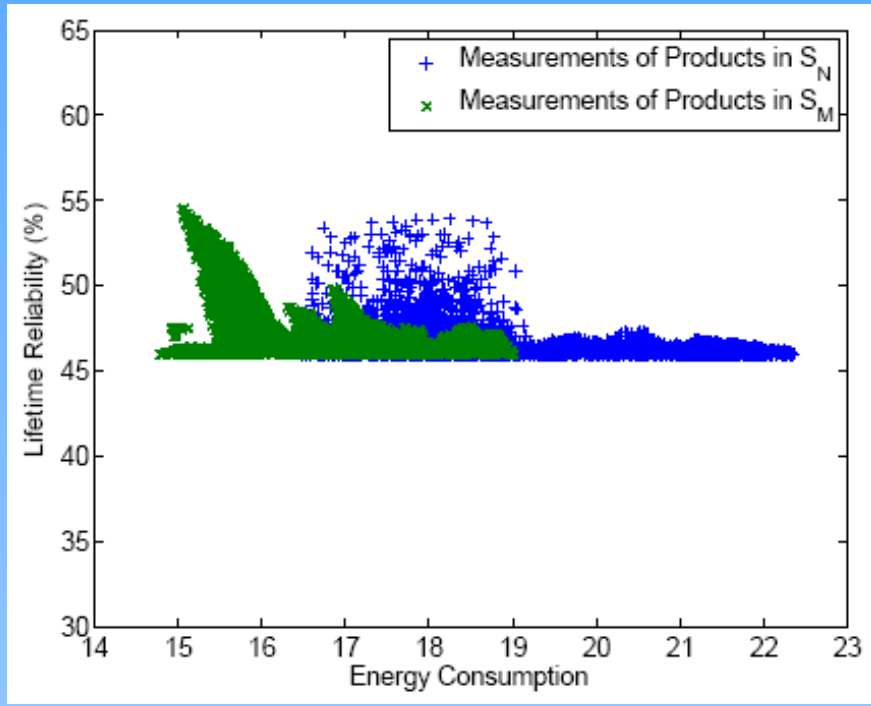
CURE
CUhk REliable Computing Laboratory

Product Personalization

- **Possible solution**
  - ✓ Generate a collection of "good" schedules instead of one at design stage
  - ✓ Optimize the design w.r.t. usage strategy distribution at design stage
  - ✓ Optimize each single product w.r.t. its unique usage strategy



Without Personalization



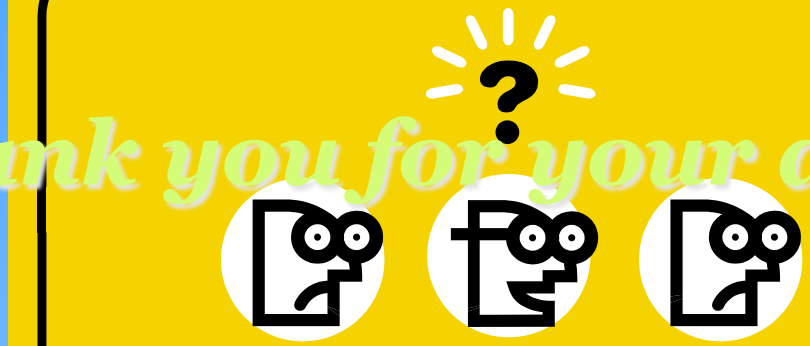
By Personalization

CUhk REliable Computing Laboratory

Conclusion

- Technology advancement has brought with ever-increasing adverse impact on lifetime reliability of multi-mode embedded MPSoC
- We propose *novel task allocation and scheduling algorithms* in this work to
  - ✓ minimize the expected energy consumption
  - ✓ meet the performance constraints
  - ✓ and meet the lifetime reliability constraint
- Experimental results demonstrate the effectiveness of the proposed methodology

Thank you for your attention !



It's **QUESTION TIME**!!

CUhk REliable Computing Laboratory