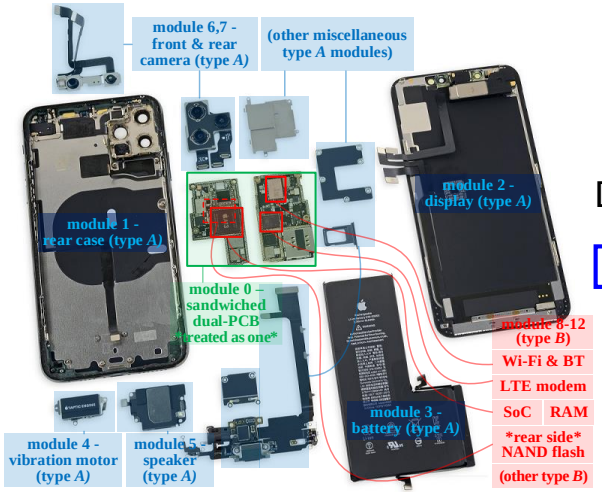
A quality status encoding scheme for PCB-based products in IoT-enabled remanufacturing

Sijie LI, You SHANG

Frontiers of Computer Science, DOI: 10.1007/s11704-020-9175-0

Problems & Ideas

- Problems of remanufacturing PCB-based products featured "*PCB-module association*", "*assembly-disassembly logic*" and "*disassembly risk*" in IoT-enabled mass production
  - How to digitalize complicated structure of PCB-based products?
  - How to connect physical product flow to digital quality information flow?
- Ideas: A quality status encoding scheme based on IoT - *IQSCT*
  - Binary-extensible sequences from 2-bit to n-bit, downwardly compatible
  - Code evolution realized on binary logic to encapsulate information



Product structure & logic

Digitalize

Connect

$$flag'_{si,j} \leftarrow \underbrace{flag_{si,0\mapsto 0b11}}_{\text{non-repairable PCB}} \cdot [\underbrace{type_{i,j\mapsto 0b10}}_{\text{type B module}} \cdot \underbrace{(flag_{si,j} \text{ and } 0b11+0b100)}_{\text{adjust quality status as well as add PCB information}} + \underbrace{type_{i,j\mapsto 0b1}}_{\text{type A module}} \cdot \underbrace{(flag_{si,j}+0b100)}_{\text{add PCB information}}] +$$

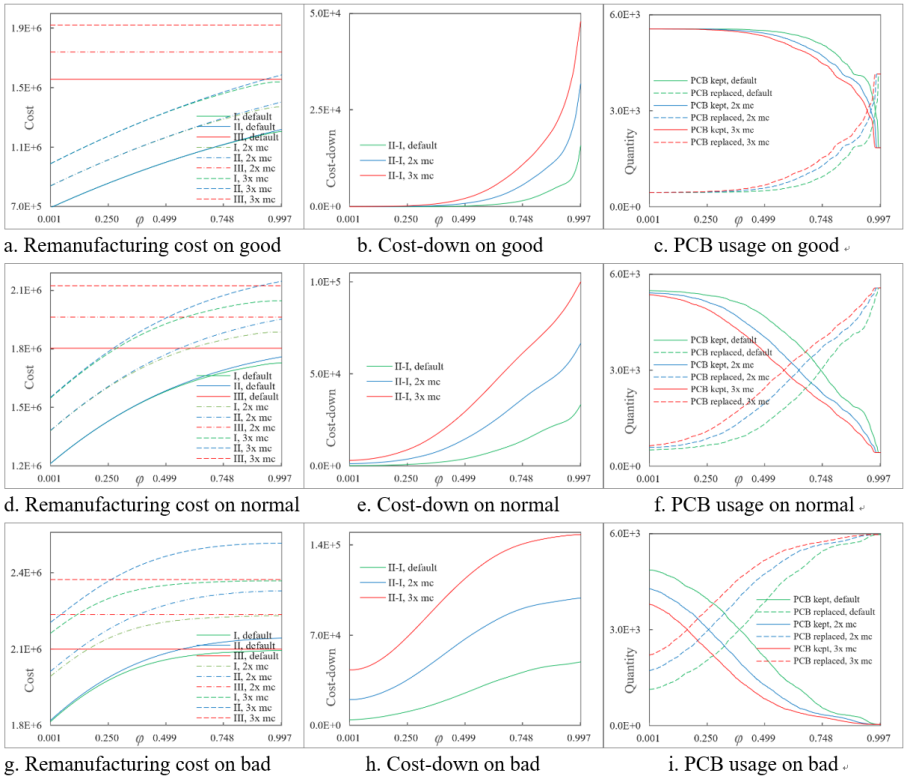
$$\underbrace{(0b1-flag_{si,0\mapsto 0b11})}_{\text{normal or repairable PCB}} \cdot \underbrace{flag_{si,j}}_{\text{keep 2-bit code}} \quad \text{where} \begin{cases} flag_{x\mapsto y} = 0b1-(flag_x \wedge y \text{ and } 0b1) \\ type_{x\mapsto y} = 0b1-(type_x \wedge y \text{ and } 0b1) \end{cases}$$

Type	PCB status	2-bit code (input)	3-bit code (output)	Comment
A	0b1, 0b10	0b0, 0b1, 0b10, 0b11	No change	/
A	0b11	0b0, 0b1, 0b10, 0b11	0b100, 0b101, 0b110, 0b111	PCB-damaged information added (indicate it from a non-repairable PCB)
B	0b1, 0b10	0b0, 0b1, 0b10, 0b11	No change	/
B	0b11	0b0	0b100	PCB-damaged information added, and missing indicator kept
B	0b11	0b1, 0b10, 0b11	0b111	PCB-damaged information added, and adjusted to non-repairable
PCB		0b1, 0b10, 0b11	/	N/A

Code evolution on Boolean algebra

Main Contributions

- Remanufacturing procedures and product information are synchronized by the proposed unambiguous measurement
- Low cost and high utilization are achieved by selective remanufacturing one-by-one based on cost expectation
- The code evolution is simple and fast to handle massive number of products
- It is well prepared for future needs, and compatible to legacy applications with original information kept



a. Remanufacturing cost on good b. Cost-down on good c. PCB usage on good

d. Remanufacturing cost on normal e. Cost-down on normal f. PCB usage on normal

g. Remanufacturing cost on bad h. Cost-down on bad i. PCB usage on bad

Case study on iPhone 11 series