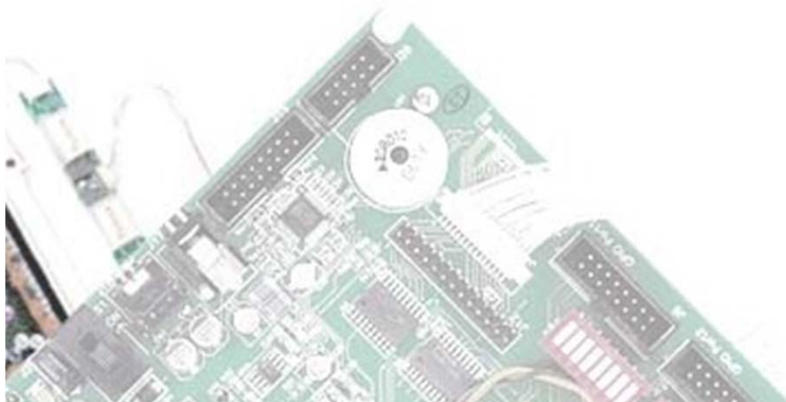
Flash Memory Reliability Model Based on Operations and Faults

April 18, 2011
Memory & Storage Architecture Lab.
School of Computer Science & Engineering
Seoul National University
JiHyuck Yun (jhyun@archi.snu.ac.kr)



서울대학교
SEOUL NATIONAL UNIVERSITY

Introduction

- Two faces of NAND flash memory

- High density
- Low access latency
- Low power consumption
- Small form factor
- High shock resistance
- Massive parallelism

⋮

FROM *THE DARK NIGHT*

- No in-place update
- Low single chip performance
- Bad block
- Power failure
- Program disturbance
- Read disturbance
- Retention error
- Endurance problem

Reliability Issues

Source: jhyoon's (jhyoon@os.snu.ac.kr) HIL slides

Introduction

- Plethora of FTLs

FAST
LAST
AFTL
DFTL
JFTL
CNFTL
Log block scheme
super-block scheme
CFTL
MS FTL
μ-FTL
Replacement block scheme
Vanilla FTL
Reconfigurable FTL
...........and so on

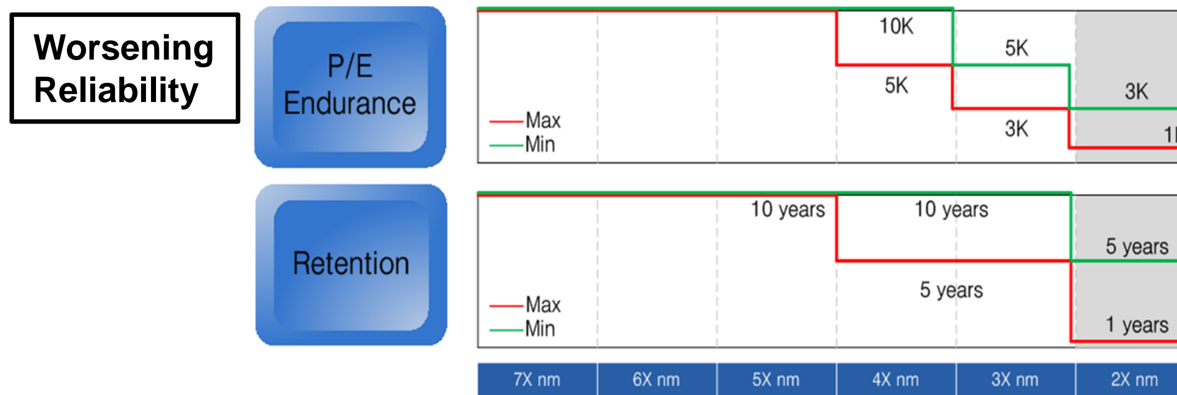
Source: yjsung's (yjsung@archi.snu.ac.kr) FTL slides

Introduction

- Key issues
  - Performance
  - Power consumption
  - Reliability
  - Cost for capacity

Much research
Interleaving, Out of order execution, Buffer management, Shadow paging
……

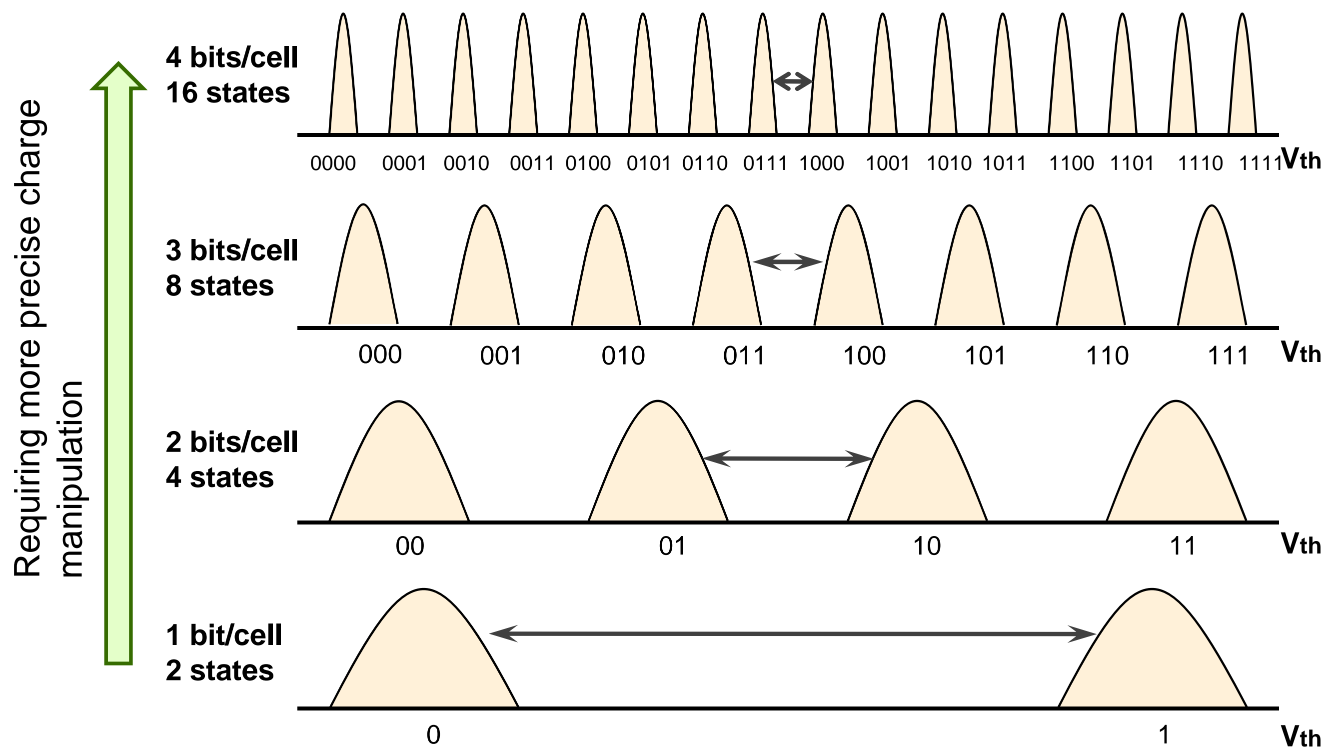
Manufacturing issues
Reduction of feature size, Multi-level cell, …



Reference: A. Niebel, "Beyond Flash Memory Technology - Defining Storage Class Memories," *2009 Non Volatile Memory conference*

Introduction

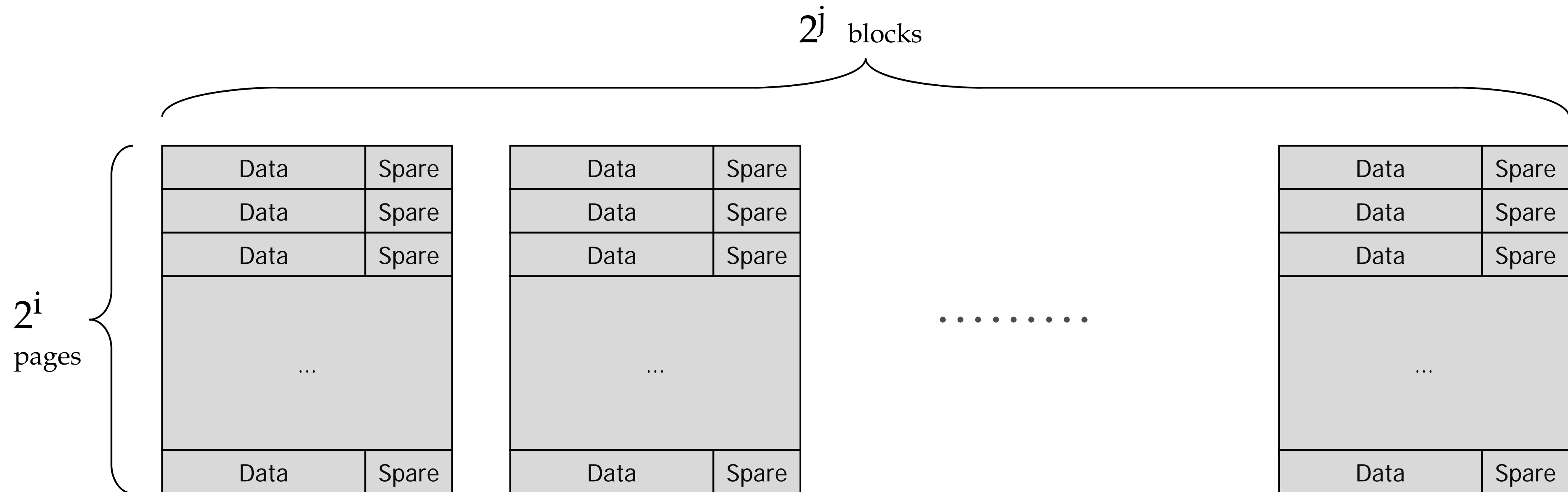
- Flash memory reliability trends
  - Reliability is getting worse as the density increases due to
    - Smaller cell size
    - MLC (Multi-Level Cell)



	SLC	MLC
Erase/Program cycle	< 100K	< 3K
ECC (per 512B)	1-bit	> 4-bit
NOP (Partial programming)	4	1

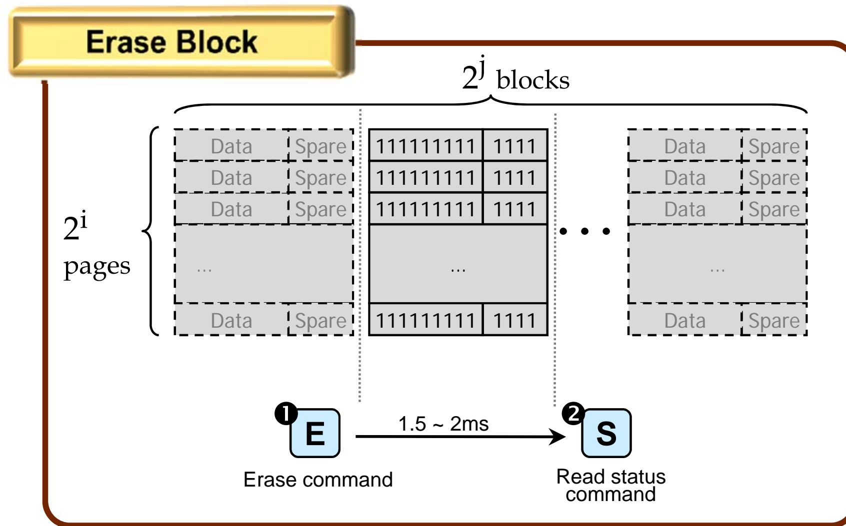
Rererence: Eli Harari (SanDisk), “NAND at Center Stage,” Flash Memory Summit 2007.

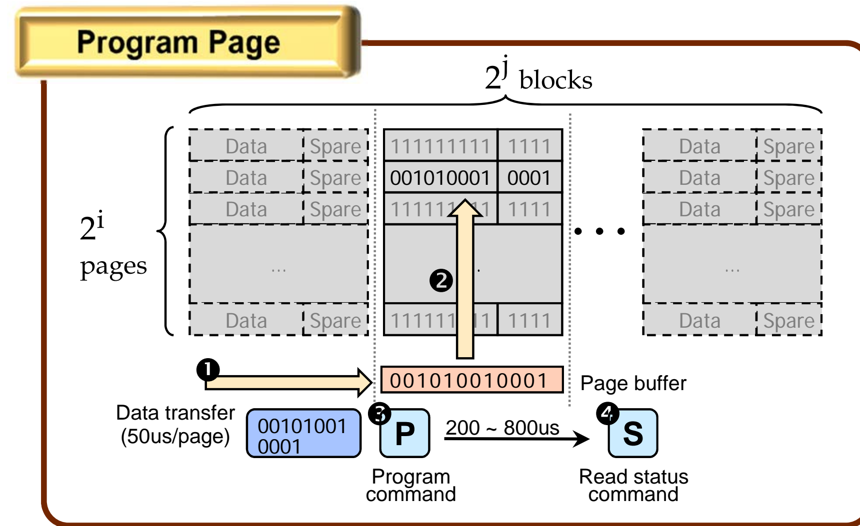
Flash memory organization

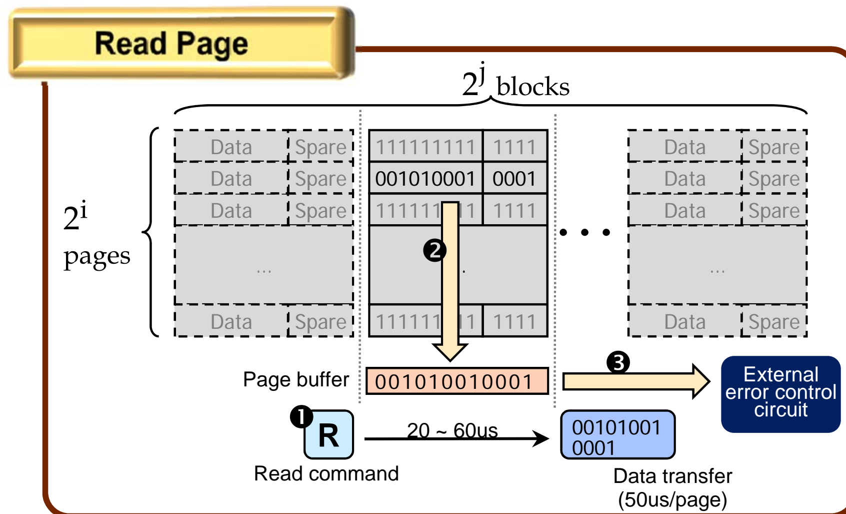


- Erase block
  - (chip#, block #)
  - 1.5 ~ 2 ms
- Program page
  - (chip#, block #, page #)
  - 200 ~ 800us
- Read page
  - (chip #, block #, page #)
  - 20 ~ 60 us

Flash memory operations







Possible failures

- Erase block : Stuck-0 failure
- Program page : Stuck-1 failure
- Read page : Bit-flipping failure
- Power failure

Causes of failures

- Over-programming
  - Due to the failure of the programming algorithm
  - Workaround: More precise steps in program-verify procedure
- Program disturbance
  - The cells not being programmed receive unintended voltage stress
  - The stressed cells appear to be weakly programmed
  - Workaround: NOP restriction, ascending order programming
- Read disturbance
  - The cells in pages not being read receive unintended voltage stress
  - The stressed cells appear to be weakly programmed
  - Workaround: External error control mechanism, Duplication, Refresh

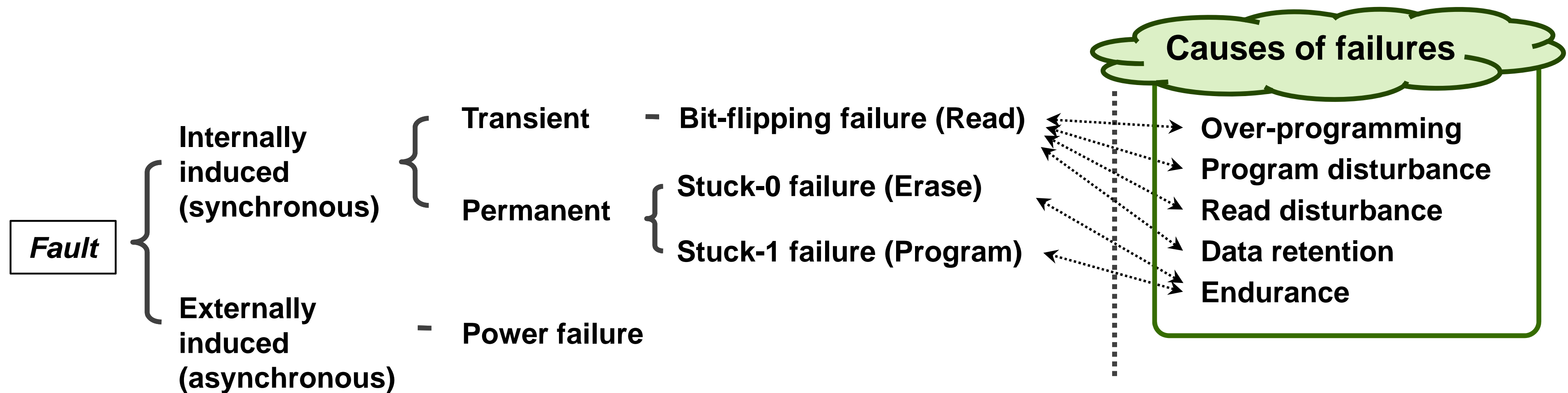
Causes of failures

- Data retention failure
  - Due to the charge loss that manifests the change of cell state
  - Shortened period
    - Current: programmed data can be retained for > 10 years
    - Future: programmed data can be retained only for a limited time (depending on the number of erase/program cycles experienced)

- Endurance
  - P/E cycles cause charges to be trapped in the dielectric that result in a *permanent shift* in cell state (not recovered by flash operaion)
  - The block that contains damaged cell is called 'bad block'
    - Manufacture-time bad block
    - Run-time bad block
      - Program failure, Erase failure

Classification of faults

- Fault types & Error control mechanism



Error control mechanism

Error Detection	Error Correction
Not Supported	Not Supported
Not Supported	Supported
Supported	Not Supported
Supported	Supported

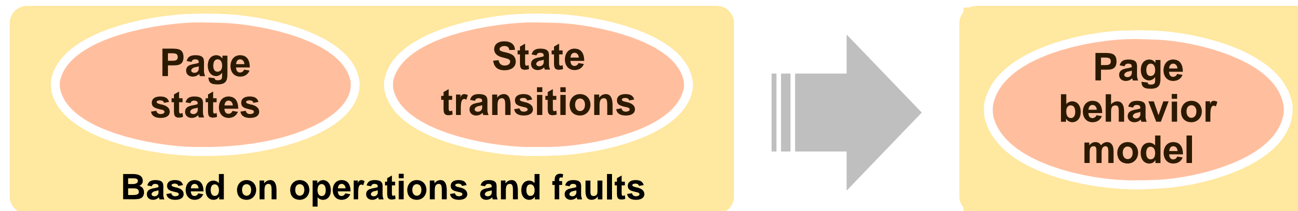
Effects of faults

- Various page states yielded by faults
  - Internal failure occurs during an erase operation in a correctly programmed page
    - Commonly, the data has not been read correctly
    - Rarely, the data has been read correctly as a result of a coincidence in the error control mechanism
  - Power failure occurs during a program operation in an erased page
    - All bits in the page are read as 1 (All 0xFF) and the page is correctly programmable, since the erased contents of the page are not affected by the program operation
    - All bits in the page are read as 1 (All 0xFF) but the page cannot guarantee the correct programming, since the page are weakly affected by the program operation
    - The data are correctly read in spite of the power failure
    - The data are not correctly read because program operation is abnormally terminated by power failure

Assumptions

- The P/E (Program/Erase) error detection circuit within the flash memory always works correctly
  - When it reports OK for an erase operation, all the bits within the block have been correctly set to 1
  - Similarly, when it reports OK for a program operation, all the bits within the page have been correctly set by the supplied data
- The external error detection is strong enough so that it is impossible for a page containing random data to be read correctly (i.e., the read operation returns ‘Data OK’) by a coincidence.

Page behavior model

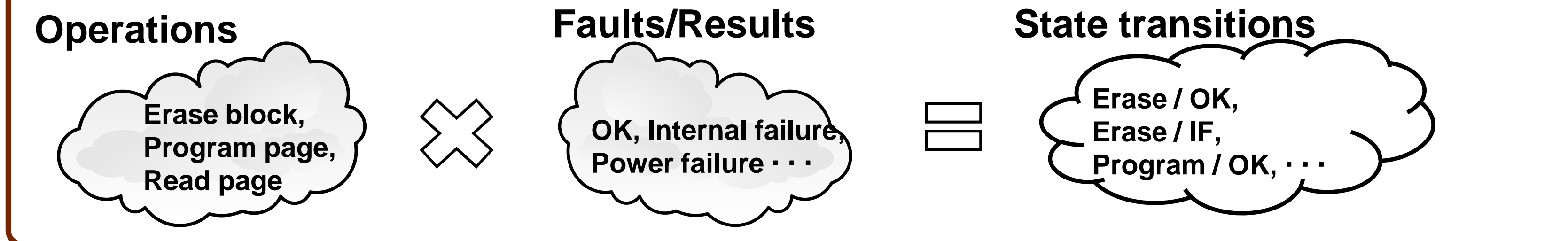


Page states

Page states
- All bits in a page are read as 1
  - Correct programming is guaranteed
  - Correct programming is *not* guaranteed
- All bits in a page are *not* read as 1
  - Data in the page are read correctly
  - Data in the page are *not* read correctly

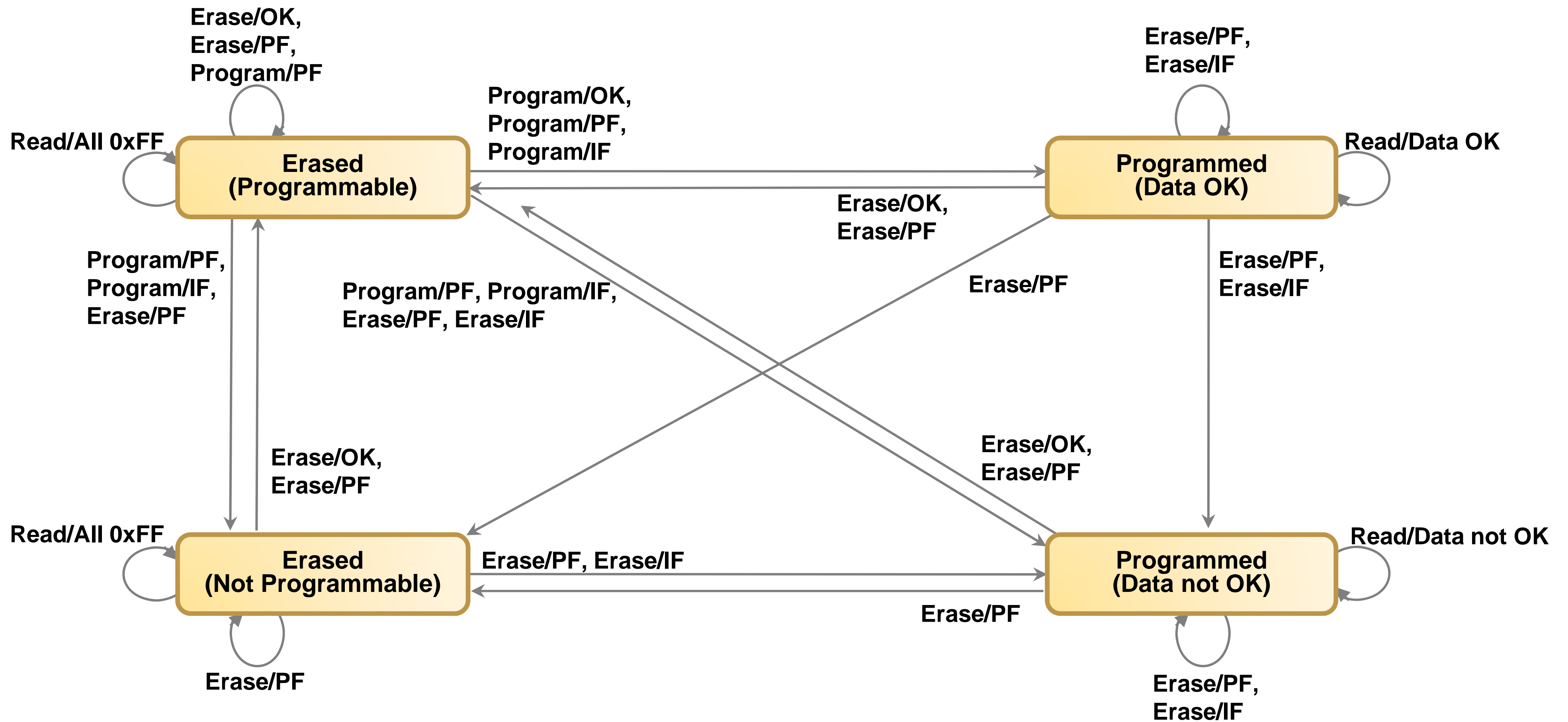
- **Descriptions of page states**
  - **Erased(Programmable)**: the block to which the page belongs has been erased successfully and a read operation will return all bits at 1 (All 0xFF)
  - **Erased(Not Programmable)**: the block to which the page belongs has been erased incompletely, but a read operation nevertheless returns all bits at 1 (All 0xFF)
  - **Programmed(Data OK)**: the page has been programmed and the data in the page are read as correct data, possibly after error detection and correction
  - **Programmed(Data not OK)**: the page has been programmed but the data in the page are read as incorrect even after error detection and correction as a consequence of internal failure or power failure

Page behavior model



- **Configurations of State transitions**
  - Combinations of flash memory operations and faults
    - Flash memory operations: Erase, Program, Read
    - Faults: IF (Internal Failure), PF (Power Failure)
    - Success results: OK, All 0xFF, Data OK, Data not OK
  - Possible combinations
    - Erase/OK, Erase/PF, Erase/IF
    - Program/OK, Program/PF, Program/IF
    - Read/Data OK, Read/Data not OK, Read/All 0xFF

Page behavior model



Page behavior model

- Validity
  - Page behavior model includes *all possible* states and state transitions except

Not allowed cases

① **Correctness cannot be guaranteed if it's allowed by system software**
ex) A page is programmed without successful preceding erase operation

② **The effect of operation differs from semantics of operation on chip-level**
ex) Previous data are read after erase operation that has already returned 'OK'

③ **Only probabilistic (not deterministic) correctness can be guaranteed if it is allowed**
ex) Incorrectly written data are corrected by erase failure and read as correct data

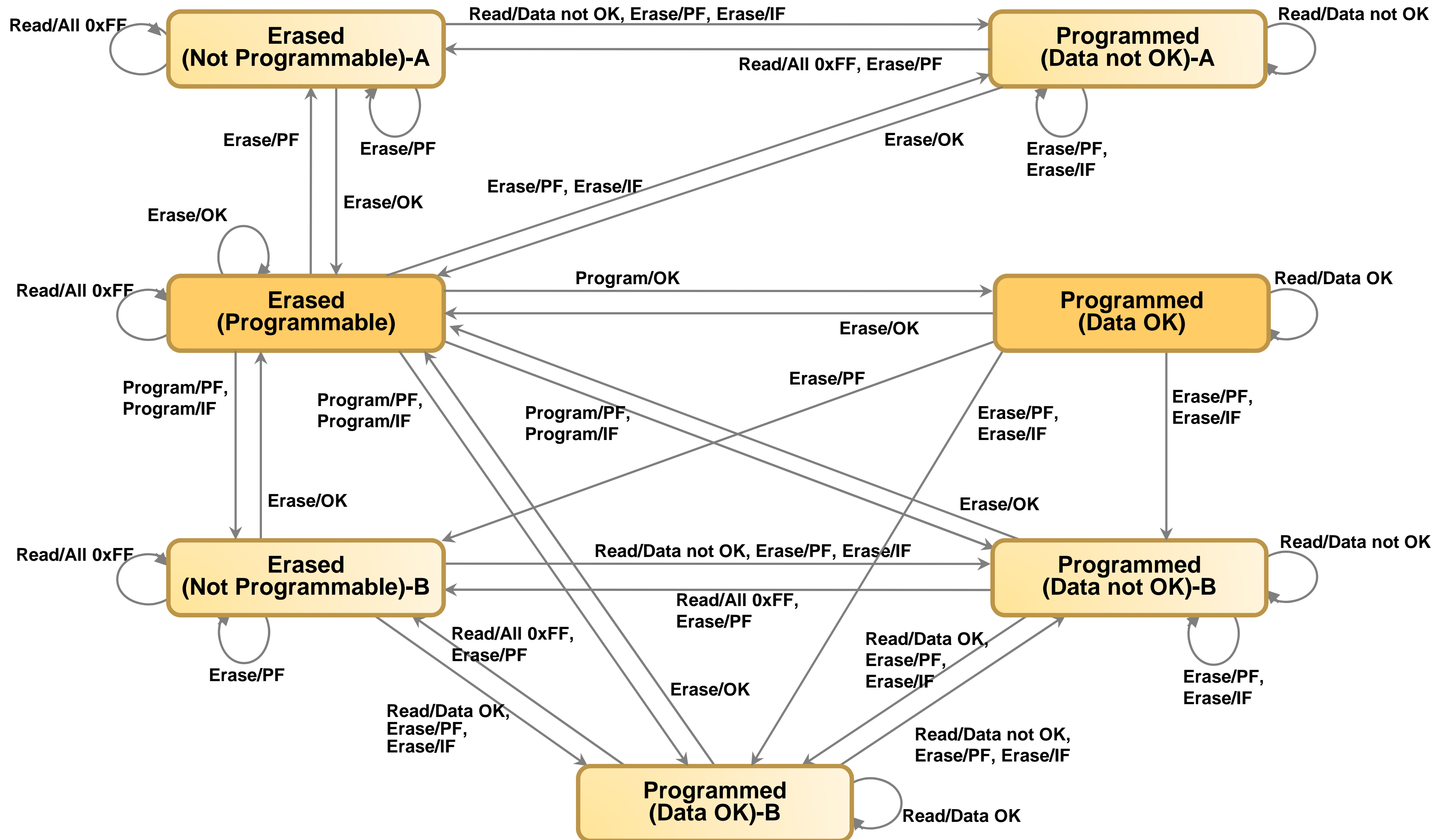
All possible transitions are allowed among states only except above cases

  - Correctness of a real-environment system is guaranteed if the system is verified based on the proposed model

Page behavior model

- Characteristics
  - Page behavior model allows all possible transitions among the states, which ensures *validity*
  - The program operations are allowed only in 'Erased (Programmable)' state, which should be also enforced by system software to guarantee the *reliability,*
  - The outcome states of erase / program operations with power failure subsume those with internal failure, which provides interrelationships between *crash recovery* and *bad block management*
  - The effect of the read operation that does not explicitly incur state transitions is *binding* the page to a specific state among the possibly multiple states yielded by program / erase operations
- Extension
  - If the failures occur during program / erase operations on a page, the page state can be changed among the possibly multiple states *nondeterministically* whenever the read operation performs on the page

Extended page behavior model



Difficulties for reliability guarantee

Aspects of power failure

<u>Power failure can occur at any time</u>

Must always pay attention to occurrence of power failure & its residual effects

<u>Power failure can occur indefinitely</u>

Must consider every successive state (maybe indefinitely)

<u>Power failure can be nested in other failures</u>

Must consider all possible states produced by nested failures

Source: jhyoon's (jhyoon@os.snu.ac.kr) HIL slides

Conclusions

- Classification of faults
  - Various types of faults in flash memory are identified and classified according to their characteristics
  - The classification implies that the synchronous and asynchronous faults can be arbitrarily nested, which causes a large of (or even unbounded) system states to be considered for guaranteeing system reliability
- Page behavior model
  - Baseline and extended page behavior models are proposed as forms of state transition diagrams that considers all possible states and state transitions with faults
  - The proposed models provide the correctness criteria ensured by the validity of the model that guarantees the correctness of a real-environments system as long as the system is verified base on the proposed models
  - The proposed models provide the inference that a page with both power failure and internal failure can be handled by crash recovery procedure which considers only power failure