

OSTEP Chapter 44

ECE 3600, Fall 2022

Table of Contents

- [1. Flash-based SSDs](#)
- [2. Flash Translation Layer \(FTL\)](#)
- [3. Garbage Collection](#)
- [4. Exercises](#)

1. Flash-based SSDs

solid-state storage device

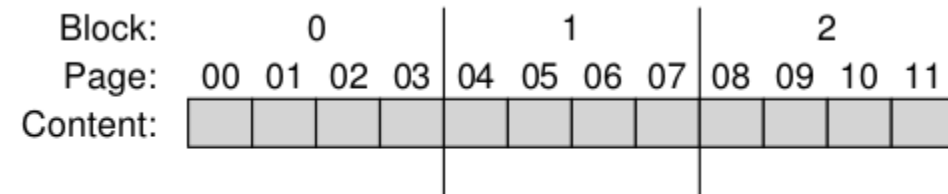


Figure 44.1: A Simple Flash Chip: Pages Within Blocks

Basic Flash Operations: read page (fast), erase block (slow), write page

block must be erased before writing any pages

rewrite page: read block, erase block, update block

		iiii	<i>Initial: pages in block are invalid (i)</i>
Erase()	→	EEEE	<i>State of pages in block set to erased (E)</i>
Program(0)	→	VEEE	<i>Program page 0; state set to valid (V)</i>
Program(0)	→	error	<i>Cannot re-program page after programming</i>
Program(1)	→	VVEE	<i>Program page 1</i>
Erase()	→	EEEE	<i>Contents erased; all pages programmable</i>

Device	Read (μ s)	Program (μ s)	Erase (μ s)
SLC	25	200-300	1500-2000
MLC	50	600-900	~3000
TLC	~75	~900-1350	~4500

Figure 44.2: Raw Flash Performance Characteristics

2. Flash Translation Layer (FTL)

map logical pages to physical blocks and pages

direct mapped - write logical 0(a),10(b),20(c) --> physical 0,10,20

```
FTL  0:0, 10:10, 20:20
Block 0          1          2
Page  0000000000 111111111 222222222
      0123456789 0123456789 0123456789
State vEEEEEEEE vEEEEEEEE vEEEEEEEE
Data  a          b          c
Live  +          +          +
```

To rewrite logical page 20: erase block 2, write page 20

log-structured - write logical 0(a),10(b),20(c) --> physical 0,1,2

```
FTL  0:0, 10:1, 20:2
Block 0          1          2
Page  0000000000 111111111 222222222
      0123456789 0123456789 0123456789
State vvEEEEEEE iiiiiiiiii iiiiiiiiii
Data  abc
Live  +++
```

To rewrite logical page 20(d): remap to free page:

```
FTL  0:0, 10:1, 20:3
Block 0          1          2
Page  0000000000 111111111 222222222
      0123456789 0123456789 0123456789
State vvvEEEEEE iiiiiiiiii iiiiiiiiii
Data  abcd
Live  ++ +
```

3. Garbage Collection

write logical 100(a1),101(a2),2000(b1),2001(b2) --> physical 0,1,2,3

Table:	100 → 0	101 → 1	2000 → 2	2001 → 3	Memory								
Block:	0				1				2				Flash Chip
Page:	00	01	02	03	04	05	06	07	08	09	10	11	
Content:	a1	a2	b1	b2									
State:	V	V	V	V	i	i	i	i	i	i	i	i	

write logical 100(c1),101(c2) --> physical 4,5

Table:	100 → 4	101 → 5	2000 → 2	2001 → 3	Memory								
Block:	0				1				2				Flash Chip
Page:	00	01	02	03	04	05	06	07	08	09	10	11	
Content:	a1	a2	b1	b2	c1	c2							
State:	V	V	V	V	V	V	E	E	i	i	i	i	

remap and erase:

Table:	100 → 4	101 → 5	2000 → 6	2001 → 7	Memory								
Block:	0				1				2				Flash Chip
Page:	00	01	02	03	04	05	06	07	08	09	10	11	
Content:					c1	c2	b1	b2					
State:	E	E	E	E	V	V	V	V	i	i	i	i	

4. Exercises

Exercises from the book using [ssd.py](#):

```
$ python ./ssd.py -s 1 -B 3 -l 30 -n 2 -q
```

```
FTL (empty)
Block 0      1      2
Page 0000000000 111111111 222222222
      0123456789 0123456789 0123456789
State iiiiiiiiii iiiiiiiiii iiiiiiiiii
Data
Live
```

```
cmd 0:: command(??) -> ??
```

```
FTL 7: 7
Block 0      1      2
Page 0000000000 111111111 222222222
      0123456789 0123456789 0123456789
State EEEEEEEvEE iiiiiiiiii iiiiiiiiii
Data      u
Live      +
```

```
cmd 1:: command(??) -> ??
```

```
FTL 7: 7 19: 19
Block 0      1      2
Page 0000000000 111111111 222222222
      0123456789 0123456789 0123456789
State EEEEEEEvEE EEEEEEEEv iiiiiiiiii
Data      u      M
Live      +      +
```

```
$ python ./ssd.py -s 1 -B 3 -l 30 -n 2 -C
```

```
FTL (empty)
Block 0      1      2
Page 0000000000 111111111 222222222
      0123456789 0123456789 0123456789
State iiiiiiiiii iiiiiiiiii iiiiiiiiii
Data
Live
```

```
cmd 0:: write(7, u) -> success
cmd 1:: write(19, M) -> success
```

```
$ python ./ssd.py -s 1 -B 3 -l 30 -n 2 -C -T log
```

```
FTL (empty)
Block 0      1      2
Page 0000000000 111111111 222222222
      0123456789 0123456789 0123456789
State iiiiiiiiii iiiiiiiiii iiiiiiiiii
Data
Live
```

```
cmd 0:: write(7, u) -> success
```

```
FTL 7: 0
Block 0      1      2
Page 0000000000 111111111 222222222
      0123456789 0123456789 0123456789
State vEEEEEEEE iiiiiiiiii iiiiiiiiii
Data u
Live +
```

```
cmd 1:: write(19, M) -> success
```

```
FTL 7: 0 19: 1
Block 0      1      2
Page 0000000000 111111111 222222222
      0123456789 0123456789 0123456789
State vvEEEEEEEE iiiiiiiiii iiiiiiiiii
Data uM
Live ++
```

```
$
```