

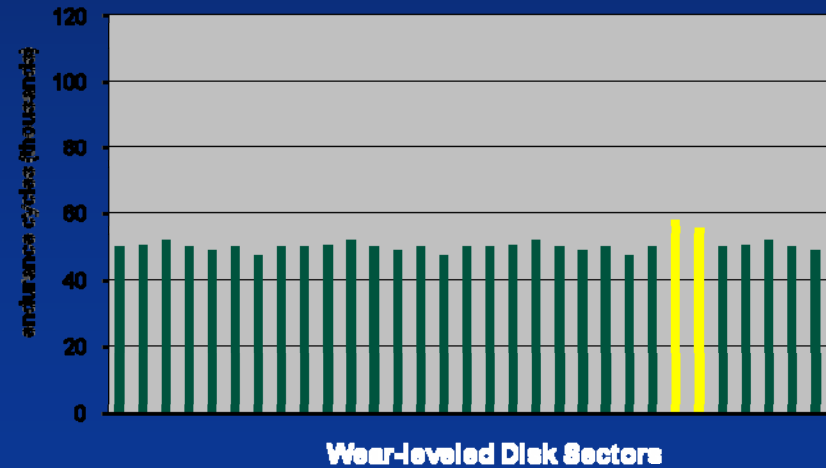
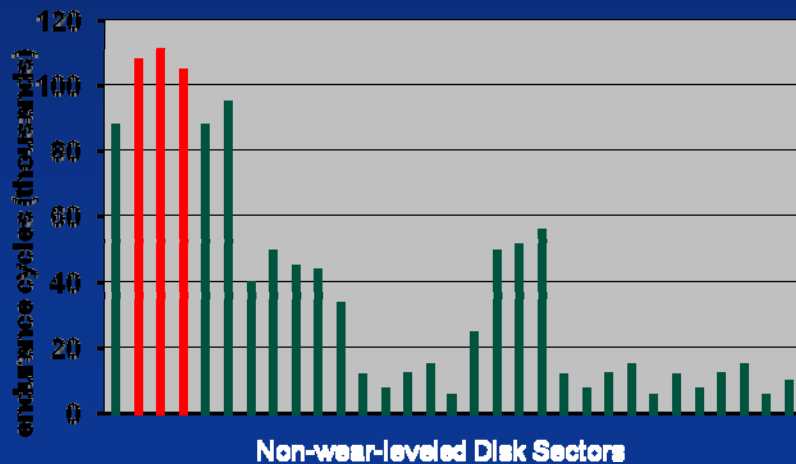
Using the Appropriate Wear Leveling to Extend Product Lifespan

Presented by: Bill Roman
Datalight Software Architect

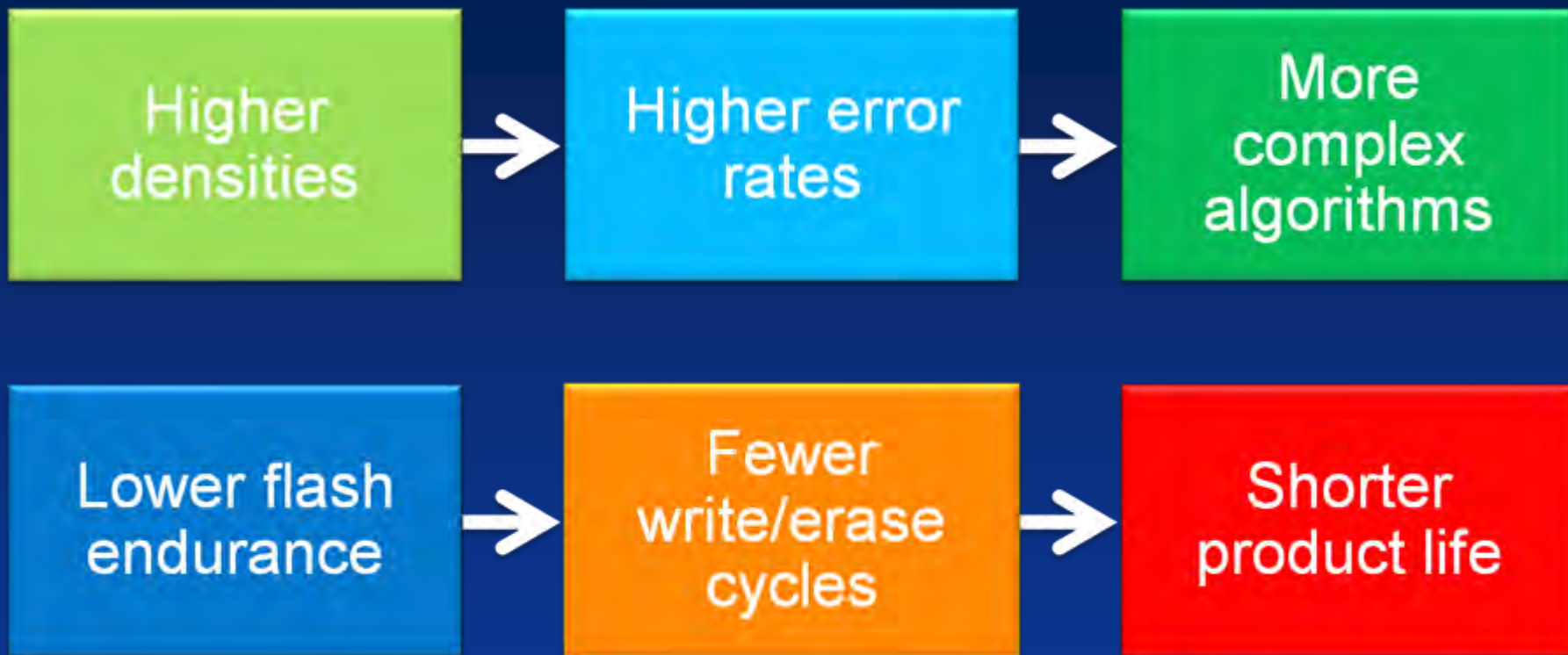
- Basics of wear leveling
- Dynamic vs. Static Wear Leveling
- Other Considerations
- Conclusions

What is Wear Leveling?

Wear leveling: a set of algorithms that attempt to maximize the lifetime of flash memory by evening out the use of individual cells.



Growing Impact on Product Life

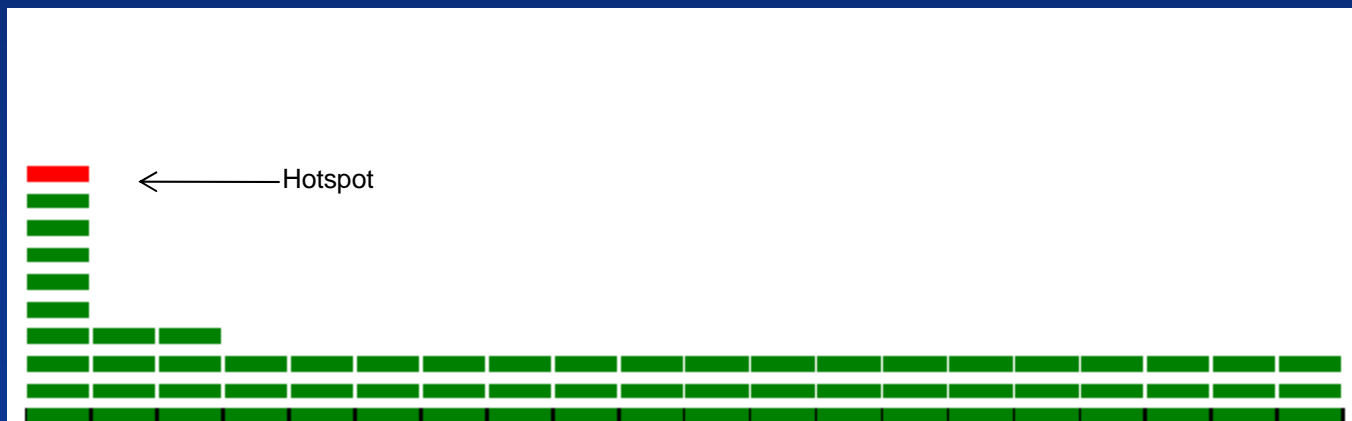


Ignorance is Costly

- Controller using read-modify-write operations
- If there is no wear leveling, an area of the flash can become unusable in just a few months
- Wear Leveling cannot be ignored, but does not need to be a 'big deal'

Conventional File System on Flash

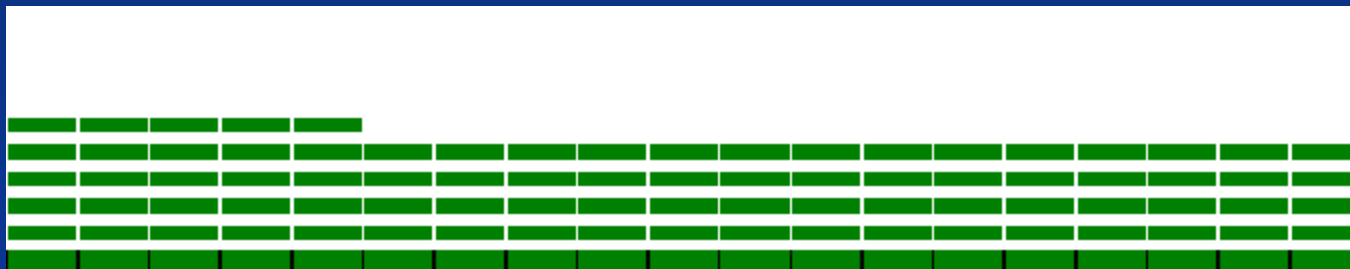
- Inefficient
- Hot spots
- Not interruption-safe
- Usable for read-only



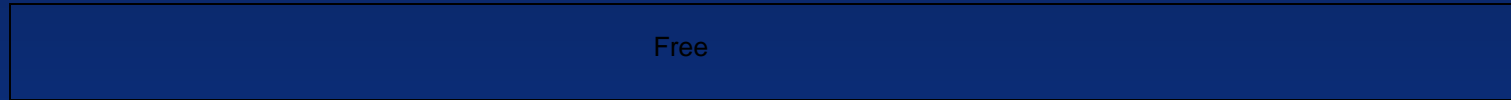
Dynamic is Usually Standard

- Largely inherent in any flash management system
 - Overwrites in flash are generally not allowed
 - Writes to NAND must be sequential
 - Requires data to be moved to efficiently use NAND

- Original JFFS used this strategy
- Strictly linear
- “Perfect” wear-leveling

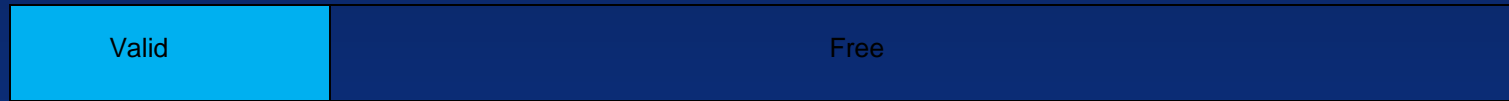


Immediately after erasing



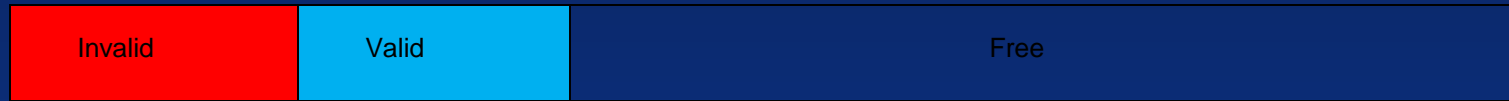
Erase Block

A file is written



Erase Block

Data in the file is modified



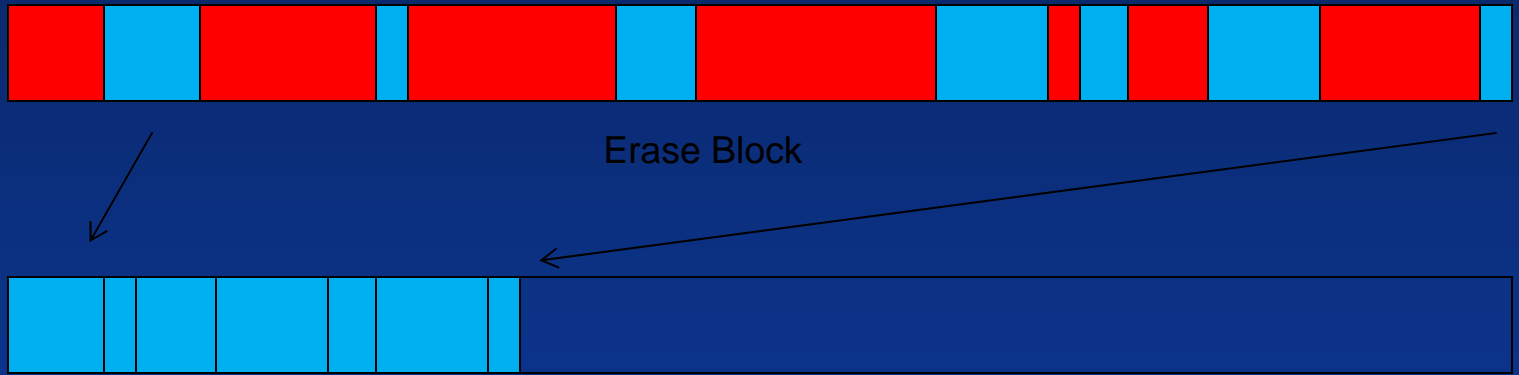
Erase Block

More files are created, modified, deleted



Erase Block

So it's necessary to compact valid data into a free erase block



FIFO



Becomes



Simple Cuts Both Ways

- Effectiveness is based on application use case
 - How much data is unchanged?
 - Only works on areas of flash which are frequently updated
 - Static data therefore reduces life of the flash

Effect of FIFO on Flash Life

- Actual Life = Specified Life * (1 - static data size/total flash size)
- Balance “perfect” wear leveling against efficiency by occasionally moving static data

Dynamic Considerations

- Ignorance is bliss
 - The system can ignore what it does not know
 - If data is never written, it is not considered (remains unmoved)
 - Decisions to move data can be based purely on performance needs

Static Does the Heavy Lifting

- Static WL is most effective
 - Largely independent of the systems' use case
 - Large (or all) portions of the media are considered
 - Static data does not significantly degrade life of the flash

Static Complicates Things

- Implementations are more complex
 - The system must track media usage and evaluate otherwise unused media
 - Requires additional erase and copy operations with zero return to performance
 - Impacts to performance can be substantial

Static and Dynamic Together

- Both approaches should be evaluated for an effective wear leveling solution.
- Dynamic is low effort and high performance
- Static ensures that media life is maximized, but at a cost to performance
- Statistical implementations may not handle certain use cases or break down entirely

Dynamic vs. Static

66	5504	17	1	103	1	57	1	75	1	60	5440	1	1	74	1
5488	1	5472	1	5235	5339	1	1	5389	1	23	5455	1	5353	1	39
27	1	52	1	161	47	1	95	1	27	81	65	1	62	1	5496
5463	1	5413	1	166	5362	1	5583	1	5387	69	1	5403	1	253	1
5534	1	5410	1	1	184	64	1	5404	1	72	3	63	5429	1	77
13	1	1	46	5469	1	1	43	1	66	1	5375	1	16	5481	1
1	136	1	23	1	169	1	5436	1	5420	5291	1	7	1	5205	5338
1	2	92	172	5431	121	5480	139	305	1	41	242	1	32	172	39
5412	5186	1	38	91	5407	88	32	126	112	5313	32	47	2	23	1
1	1	1	837	1	1	1	1	87	5355	5359	1	1	4	1	1
1	155	5352	1	1	1	1	40	1	1	1	1	99	1	1	1
2	39	1	86	5265	1	111	1	26	5401	1	1	35	1	38	1
22	51	1	5291	1	32	5490	1	55	1	40	1	100	1	52	1
70	5404	1	554	1	5264	5374	1	121	1	5321	5348	1	84	1	115
5457	1	1	147	1	5347	1	312	1	5369	230	1	1	42	1	104
1	5332	1	5398	140	1	5418	1	5619	1	1	177	4	1	5424	1
246	1	288	1	1	5466	82	1	2	1	5339	29	1	5451	1	144
5377	1	5318	1	154	5097	1	10	1	5314	5371	1	1	5462	1	243
169	1	116	1	5364	20	1	3	1	14	364	238	1	5330	64	1
5328	1	5384	5410	1	1	5350	1	5371	1	191	1	5318	5535	1	1
61	1	61	1	5348	5521	1	143	1	128	1	5339	1	1	5391	323
1	5396	1	61	1	5371	5342	5486	5428	1	5495	1	114	5305	1	5310
1	5491	42	1	60	1	89	44	1	5438	1	1	5357	2	1	67
1	5328	5383	1	5431	1	48	37	1	5460	1	5336	1378	1	8	1
65	1	5427	1	2	1	1	5366	105	1	15	1	9	17	79	192
1	78	77	1	5339	1	179	1	5387	1	3	5347	1	1	133	1
26	1	5455	65	33	5401	1	5446	1	69	1	331	57	1	40	1
10	35	1	3324	1	60	3319	1	70	1	32	72	214	1	5511	1
55	5379	117	73	53	2	5436	1	737	1	5488	135	1	5405	1	1
183	85	1	167	1	103	1	57	1	1	61	123	1	5	5396	5399
62	5564	73	114	1	18	5257	74	93	80	1	90	99	5386	5404	5402

Dynamic Only

2252	2163	1573	1508	1944	2167	1869	1590	2261	2188	2233	1922	2182	1768	1601	1598
2171	1610	1912	2058	1598	1643	1753	2211	2042	2151	1597	1554	1624	2139	1541	1756
2302	1648	1542	1586	1683	2294	1589	1722	1673	2167	1991	1579	2333	2208	1573	2055
1702	1763	1524	1944	2197	1656	1922	1544	1664	1551	1541	2174	2191	1889	2193	2168
1812	2230	2184	2206	1707	2124	2163	1531	2184	2164	1988	1658	1591	1613	2225	2225
2167	1549	1548	1669	1593	2000	2234	2242	1616	2149	2213	2228	2224	1592	1539	1719
1530	1542	2217	1673	1572	1717	1542	1536	2190	1614	1634	1631	2221	1512	1578	1509
2161	1673	2259	1563	1703	2038	2245	2255	1722	2248	1575	1524	2249	2207	1666	2181
2194	1601	2240	1822	2163	1526	2204	2163	1687	2256	1593	2146	1627	1950	2139	2181
1884	1536	2185	2289	2117	2240	1851	1510	2292	1937	1568	1623	2308	2178	1568	2154
1638	2176	1703	1625	1608	1520	2142	2200	2161	2185	2214	1774	1590	2086	1529	2159
2164	2130	1521	2162	1651	2246	1620	1823	1871	1511	2329	1639	1522	1566	2175	1675
1540	1736	1696	1513	2249	2366	1909	1517	2194	1608	2165	1628	2216	1920	1905	2154
1652	2198	1591	1543	1843	2205	1549	1533	2150	1553	1580	2165	1522	1568	1850	1648
1776	1872	2026	2253	1532	2216	2156	1540	1505	1607	2260	2306	1612	1544	1771	1629
2250	1506	1579	1578	2037	2143	1668	1512	2250	2201	1801	1648	2197	1568	2158	1859
1654	1676	2248	2261	1666	1596	1981	2034	1545	2268	2308	2249	2292	1526	1530	2160
1607	1880	1537	1904	2236	1522	2216	2220	2162	2213	2174	2190	2193	2195	2222	1509
2258	2194	2134	1523	1720	2254	2257	1525	1613	2194	1970	1530	2102	2219	1944	1939
2061	2131	1855	2202	2303	2242	1912	2184	2289	1844	1533	2188	2232	1697	2044	2213
2406	2072	1558	1519	2181	1505	1716	1983	1558	2233	1606	1596	1509	1657	2232	2255
2208	2267	2229	2259	2248	2191	2005	2154	1539	2199	2249	2307	2197	1797	2375	1967
2173	2219	2192	1693	2109	1596	2147	1550	1639	2094	2177	2285	1549	2395	1948	1886
1555	2206	2206	2213	2350	2064	2053	2152	2286	2131	1621	1528	1778	1912	1513	2257
2190	2222	2245	1513	1548	1833	1622	1592	2252	1780	1572	2180	1696	2216	1602	2248
1871	2189	2185	1636	1821	2138	1611	1542	2197	2338	1625	2129	2184	2211	2217	1572
2146	1531	1559	1546	1683	1607	1519	1568	1975	1951	1602	2189	1550	1524	1562	2165
2060	1581	1657	2170	1622	2162	2284	1626	1544	1506	1639	1606	2143	2286	1614	2185
1721	1936	2312	1867	2147	2254	1774	2306	2175	1669	2177	1548	1579	2209	1926	1630
1756	2083	1863	1587	1520	1530	1821	1572	2225	2256	1923	1913	1676	2235	1534	1540
1548	2168	1594	2218	1534	1589	2176	1605	2216	1506	1530	1516	2200	1518	2212	1728

Static + Dynamic

Do I really need to care about Wear Leveling?

- Use case description
 - System is a 40GB SSD in a notebook running Windows XP
 - OS and applications account for 18GB and 12GB respectively
 - Use is primarily communication, presentations, etc.
 - The flash used consists of MLC parts with 2.5K cycle rating

- Average daily writes by the OS... about 1.8GB... Surprised?
- User application writes are dwarfed by comparison
- A single cycle of the entire media requires nearly a month (22 days)

Use Case Static Wear Leveling

- And assuming reasonably effective static wear leveling (60%), entire media is available for wear leveling
- $2,500 \text{ cycles} \times 22 \text{ days} = 1,855 \text{ months} \times .60 = 211 \text{ years}$

Use Case Dynamic Wear Leveling

- Assuming we have a lackluster dynamic algorithm (90%) reduce available size by 30GB (application and OS)
- $2,500 \text{ cycles} \times 6 \text{ days} = 750 \text{ months} \times 0.6 = 37 \text{ years}$

What Wear Leveling Does for Reads

- Dynamic wear leveling will not move areas that are only read
- High differences in erase counts result in higher BER
- Uncorrectable error rates are increased by 2-3 orders of magnitude

Wear Leveling for Reads

- Reads ~ Writes, but startup and hibernate cost additional 7GB of reads
- NAND manufacturers recommend cycling after 100,000 reads
- In use case, upwards of 2000 reads from the same areas daily
- With no caching and no wear leveling, cycle limits reached in 2.5 months

Balancing Wear Leveling with Performance

- Wear level operations should be bounded
 - Impacts to performance must be low
 - Take advantage of idle time if possible
- Features such as trim, pre-erase, discards will mitigate negative performance impacts of wear leveling

Balancing Wear Leveling with Performance

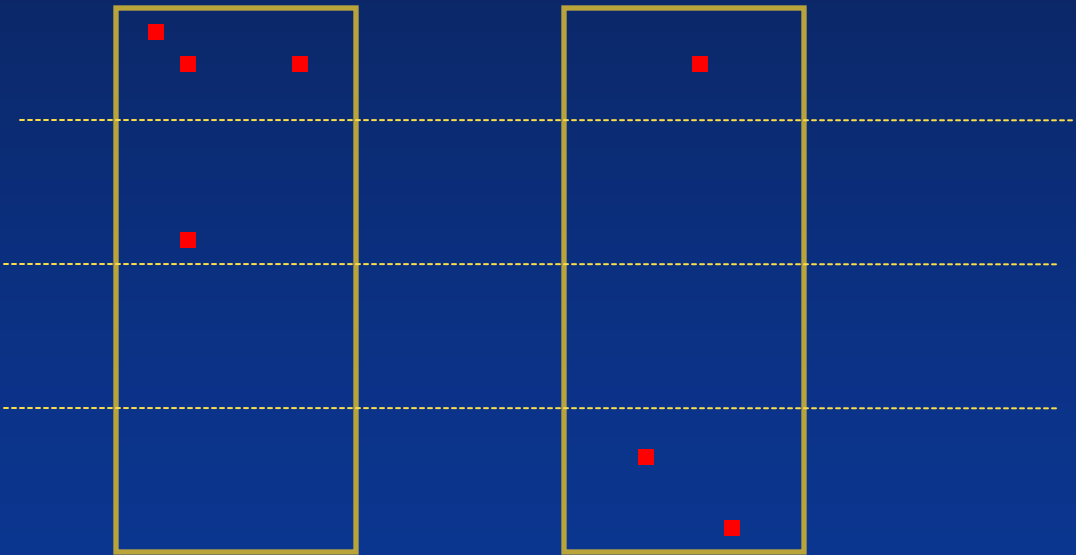
- Interleaving = speed
 - We interleave multiple plans on single devices for concurrency
 - Then two or more devices for a wider data path (more concurrency)
 - And then we do it again with multiple NAND channels or banks
 - And then once again in the field (RAID)

Definition of Interleaving

- To arrange data in a non-contiguous way to improve performance – Webopedia.com

Interleaving

Chip 1



Chip 0

Interleaving can cause unacceptable error counts

Interleaving = More Errors

- Interleaving two NAND devices can double the number of bad blocks
- Requiring more complex systems, more overhead, and/or stronger EDC

- Wear leveling requirements are *substantially* dependent upon use case
- Impacts to performance will increase with more static data in the disk
- Embedded systems will have more stringent requirements
- Understand your target customers' use case

References

- <http://ieeexplore.ieee.org/Xplore/login.jsp?url=http://ieeexplore.ieee.org/iel5/4550747/4558854/04558857.pdf&authDecision=-203>
- <http://www.stanford.edu/class/ee380/Abstracts/081112-Fazio-slides.pdf>
- http://download.micron.com/pdf/presentations/events/WinHEC_Cooke.pdf