Jak działają pamięci masowe

Niemal każdy z nas jest użytkownikiem jakiegoś rodzaju pamięci masowej zbudowanej z zastosowaniem układów NAND Flash. Są one powszechne i pracują jako wbudowane w urządzenie dyski twarde wyposażone w interfejs SATA lub PCIe lub pamięci wymienne, takie jak: karty SD, uSD, MMC, eMMC, pamięci ze złączem USB itp. Mimo powszechności niewiele osób zna sposób funkcjonowania takich pamięci – zadaniem tego artykułu jest jego przybliżenie.

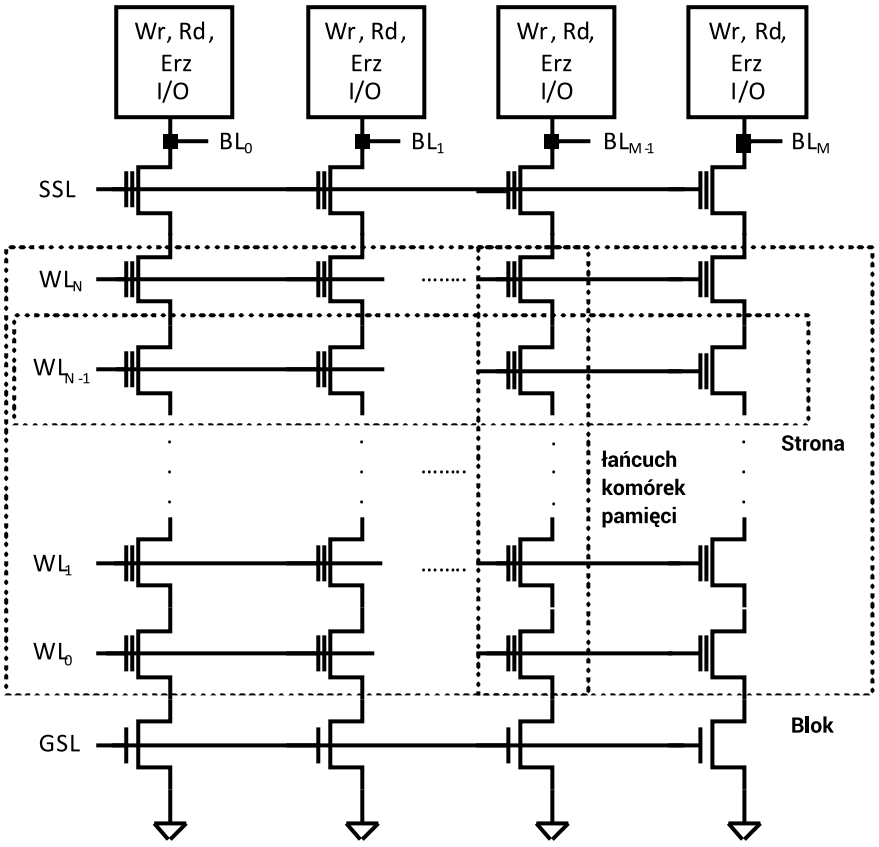
Pamięci masowe z punktu widzenia użytkownika po prostu przechowują one pliki, natomiast z technicznego punktu widzenia, przechowują one dane w sektorach typowo wielkości 512 B. Można powiedzieć, że MSD są „adresowane sektorami".

W NAND Flash, które stanowią pamięć danych, funkcjonuje zupełnie inny schemat adresowania niż w MSD. Dane są zapisywane i odczytywane stronami o wielkości 2…16 kB, a zapis wymaga uprzedniego wykasowania bloku zawierającego setki stron. Cechy te sprawiają, że w MSD występuje pewne wzmocnienie zapisu (WAF –*Write Amplification Factor*), uzależnione od aplikacji użytkowania. Ponadto, w niektórych zastosowaniach brak wsparcia dla komendy TRIM powoduje tzw. „*garbage collection*".

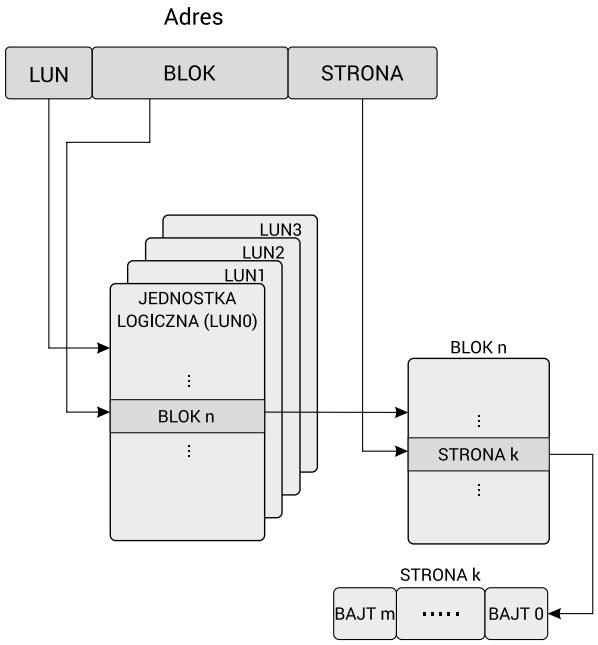
W artykule skupiamy się na poruszonych w wstępie zagadnieniach i omawiamy je w kontekście budowy pamięci NAND Flash oraz MSD. Dzięki wielu odnośnikom do standardów i specyfikacji, czytelnik może poszerzyć swoją wiedzę w prezentowanej dziedzinie.

Ze względu na niską cenę i dużą pojemność, pamięci NAND Flash (pat. US20110128788 A1) są powszechnie stosowane jako pamięć danych w MSD. Duża pojemność i niska cena wynikają z ich budowy. Komórki pamięci połączone są w łańcuchy, kilkanaście – kilkadziesiąt tysięcy łańcuchów tworzy stronę. Kilkadziesiąt do kilkaset stron tworzy blok **(rysunek 1).** W tak utworzonym bloku, komórki pamięci znajdują się na przecięciach linii kolumn BL (*Bit Line*) oraz wierszy WL (*Word Line*), dzięki czemu jest możliwe ich duże upakowanie na relatywnie niewielkim obszarze płytki krzemowej. Innym czynnikiem mającym wpływ na cenę i pojemność jest możliwość zapisu wielu bitów informacji w pojedynczej komórce pamięci tj. SLC – 1 bit, MLC – 2 bity, TLC – 3 bity (pat. US 7443724 B2).

Budowa układu pamięci determinuje sposób adresowania danych (**rysunek 2**). Więcej na ten temat można znaleźć w dokumencie JESD230B, ONFI spec. 4.0. We współczesnych NAND Flash najmniejszą jednostką pojemności podlegającej programowaniu i odczytowi jest strona o pojemności 2…16 kB. Przed ponownym programowaniem zapisanej strony, należy wykasować wszystkie strony znajdujące się w bloku. Dla przykładu, jeżeli blok ma 256 stron o pojemności 16 kB, to pojedyncza komenda ERASE kasuje blok mieszczący 4 MB danych. Załóżmy scenariusz, w którym zapisujemy tylko jeden 512 bajtowy sektor dysku, następnie wydajemy komendę ATA FLUSH CACHE (spec. T13/1699-D). W najlepszym przypadku zostanie zaprogramowana jedna strona NAND Flash o przykładowej pojemności 16 kB. W ten sposób, w NAND Flash jest programowany obszar 31 razy większy niż wynika to z faktycznych potrzeb – w pewnym uproszczeniu, właśnie ten brak dopasowania rozmiaru sektora do pojemności strony jest odpowiedzialny za powstawanie wzmocnienia zapisu WAF (doc. JESD218).



Rysunek 1. Typowa architektura pamięci NAND Flash



Rysunek 2. Podział przestrzeni adresowej NAND Flash

Rysunek 3. PCBA

W tym momencie zauważamy, że pamięć masowa nie może działać bez mechanizmu translacji adresów, którymi posługuje się komputer (adres logiczny) na adres bloku/strony NAND Flash (adres fizyczny). Ten mechanizm nosi nazwę FTL (*Flash Translation Layer*, doc. ISSN 2250-3153) i jest on odpowiedzialny za translację adresów oraz wykluczanie z użytkowania niezdatnych bloków pamięci. W FTL na ogół jest implementowany algorytm zapewniający równomierne obciążenie cyklami zapisu/kasowania (p/e) wszystkich bloków NAND Flash (*Wear Leveling*, doc. TN-29-42). Jest on konieczny, aby wielokrotnie powtarzany zapis sektora (pliku) nie powodował przekroczenia maksymalnej liczby cykli p/e statycznie zmapowanej strony.

Konsekwencją implementacji mechanizmu *wear leveling* jest skłonność MSD do przenoszenia nieaktualnych danych. Wyobraźmy sobie, że w systemie operacyjnym nie jest zaimplementowana komenda TRIM (spec. T13/1699-D, systemy starsze od Windows 7 i Linux kernel 2.4). W takiej sytuacji, system nie informuje SSD, które sektory zostały zwolnione przez usunięte pliki i które można usunąć. Późniejszy wewnętrzny proces defragmentacji przenosi sektory zassane nieaktualnymi danymi w inne obszary NAND Flash zwiększając WAF.

Rozważany przypadek użycia TRIM dotyczy SSD. Karty SD/uSD/MMC mają równoważne polecenie „ERASE" (eMMC również TRIM), które umożliwia wykasowanie grupy sektorów (doc. JESD84, doc. SD Specifications Part 1 Physical Layer Simplified Specification). Gdy system ma możliwość bezpośredniego operowania na interfejsie karty, aplikacja powinna zadbać o kasowanie zwalnianych sektorów. Pamięci typu USB są zupełnie pozbawione tej możliwości. Jeszcze inną możliwość obniżenia WAF daje buforowanie cykli zapisu w pamięci RAM (doc. BPLRU: A buffer management scheme for improving random writes in flash storage). Każdy kontroler MSD ma wbudowaną lub zewnętrznej pamięć RAM. Jest ona używana w celu skompletowania danych, możliwie dobrze wypełniających całą pojemność strony NAND Flash. Zatem buforowanie cykli zapisu w pamięci RAM systemu operacyjnego pozwala na zwiększenie sprawności wbudowanych mechanizmów optymalizacji.

Pokrótce przedstawione mechanizmy są zaimplementowane w kontrolerze MSD, głównie w jego oprogramowaniu. Każdy kontroler składa się z interfejsu do komunikacji z hostem, mikroprocesora, pamięci RAM, ROM, kontrolera pamięci Flash oraz opcjonalnie – kontrolera DMA i pamięci DRAM. Przykładowo, kontroler PS2251-07 pamięci typu USB 3.0 (**fotografia 3**) ma dwukanałowy kontroler NAND Flash z interfejsem synchronicznym i asynchronicznym (doc. JESD230B), z detekcją/korekcją błędów kodami BCH (Mathematisches Institut der Universitat: Extending and lengthening BCH-codes), mikroprocesor oraz regulator napięcia zasilania NAND Flash. Kontroler ten nie wymaga rezonatora kwarcowego w celu synchronizacji z interfejsem USB hosta. Budowa kontrolerów kart SD/MMC (np. PS8036/PS8223, **fotografia 4**) jest zbliżona do kontrolera USB, od którego głównie różnią się interfejsem. Ich interfejsy zapewniają komunikację zgodnie z standardami, odpowiednio: SD 4.2, eMMC 5.0, umożliwiając ich stosowanie w niemal każdej aplikacji wbudowanej, na przykład, dysk systemowy lub pamięć danych.

Z grupy omawianych, najbardziej rozbudowany jest kontroler dysku SSD. Przykładowy PS3110 wyposażono w aż 4 mikroprocesory, dzięki czemu jest możliwe uzyskanie dużej liczby operacji na sekundę IOPS (I/O Operation Per Second). W jego strukturze znajdziemy również kontroler pamięci DRAM, która służy jako pamięć metadanych kontrolera oraz pamięć operacyjna i zaawansowany kontroler DMA. Dzięki tym rozwiązaniom, kontroler ten pozwala na zapis/odczyt danych z szybkością 560 MB/s.

Dzięki zaawansowanym rozwiązaniom wbudowanym w omawiane kontrolery, użytkownik nie musi poświęcać czasu na implementacje własnych mechanizmów optymalizacji, translacji adresów i innych. Należy jednak pamiętać, że budowa MSD musi być dopasowana do aplikacji. Czytelników zachęcamy do zapoznania się z wskazaną literaturą, która w pozwoli na dogłębne zapoznanie się z budową pamięci masowych.

WILK ELEKTRONIK S.A.

Rysunek 4. CFast GOODRAM Industrial