Pamięć podręczna (ang. *cache*) i jej wykorzystanie przy optymalizacji programów równoległych.

Rafał Lenart

15 lipca 2025

Outline

Jaka pamięć?

Jaka powinna być pamięć operacyjna?

Jaka pamięć?

Jaka powinna być pamięć operacyjna?

1. Szybka

Jaka pamięć?

Jaka powinna być pamięć operacyjna?

1. Szybka
2. Stabilna

Jaka pamięć?

Jaka powinna być pamięć operacyjna?

1. Szybka
2. Stabilna
3. Pojemna

Jaka pamięć?

Jaka powinna być pamięć operacyjna?

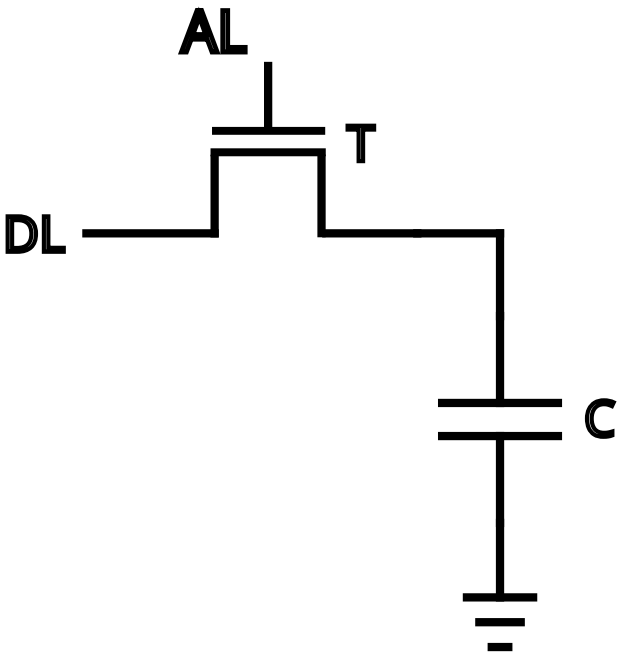
1. Szybka
2. Stabilna
3. Pojemna
4. Tania

Rodzaje pamięci RAM

Pamięć ram dzieli się na dwa rodzaje.

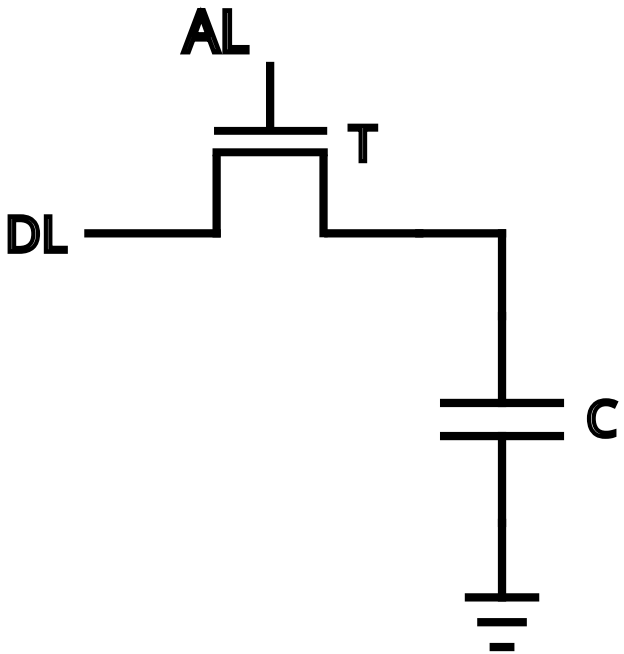
1. **DRAM** - Dynamic RAM
2. **SRAM** - Static RAM

1 bit pamięci DRAM



- *AL* - linia adresująca
- *DL* - linia danych
- *T* - tranzystor
- *C* - kondensator

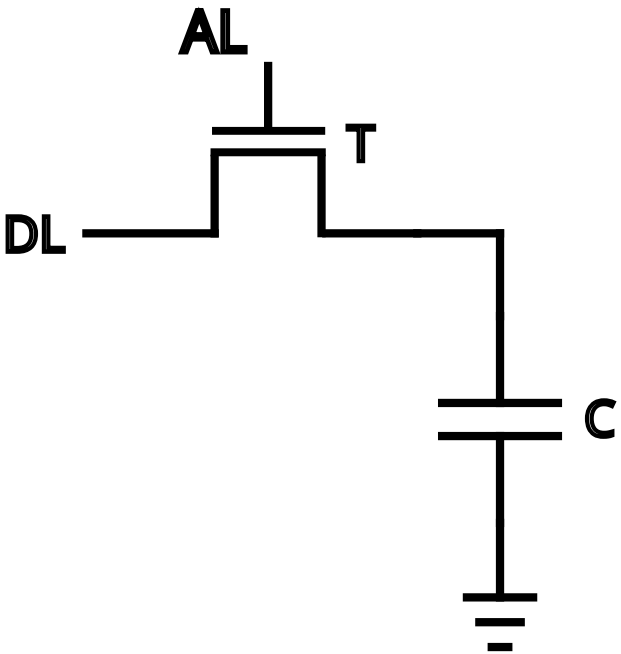
1 bit pamięci DRAM



- *AL* - linia adresująca
- *DL* - linia danych
- *T* - tranzystor
- *C* - kondensator

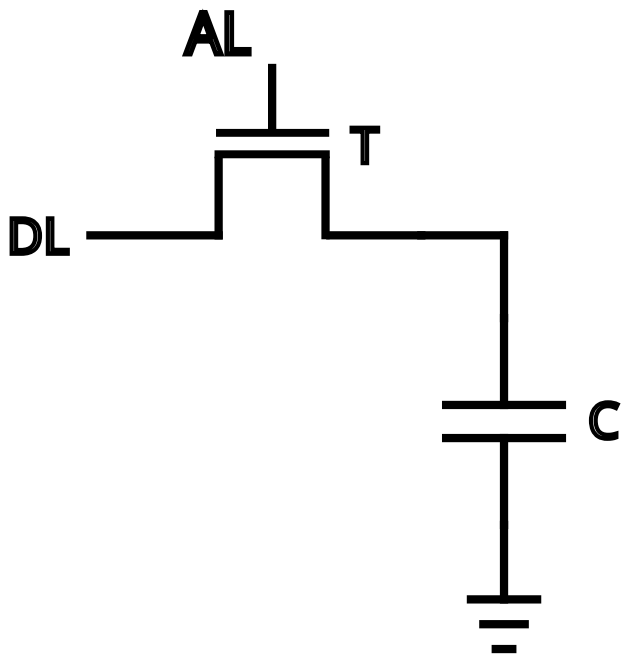
Co tu jest problemem?

Cechy DRAM



Cechy DRAM:

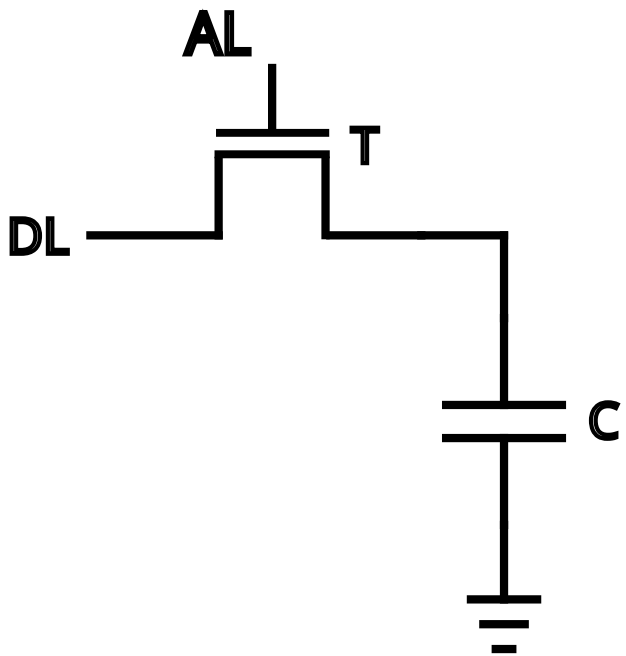
Cechy DRAM



Cechy DRAM:

- Tylko jeden tranzystor na bit.

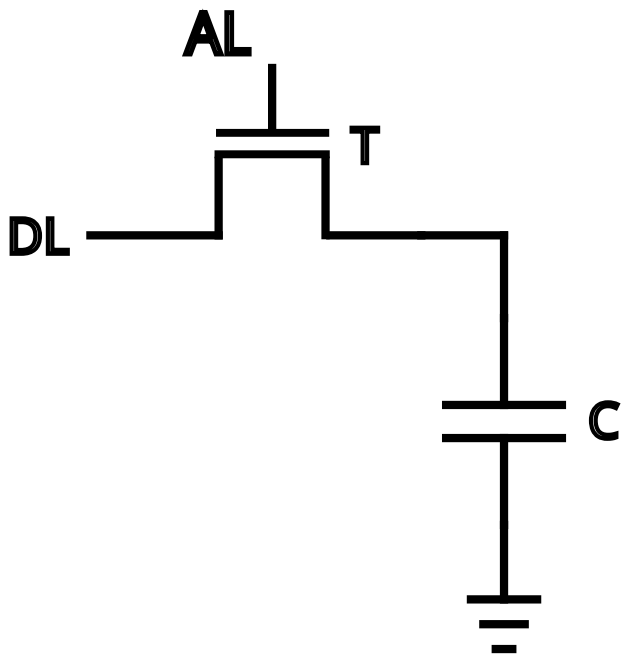
Cechy DRAM



Cechy DRAM:

- Tylko jeden tranzystor na bit.
- Potrzebuje odświeżania ("wyciekający" ładunek).

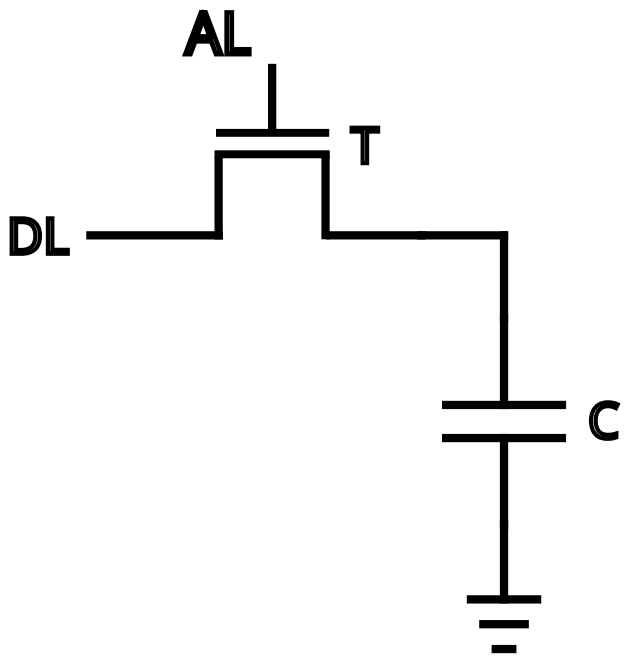
Cechy DRAM



Cechy DRAM:

- Tylko jeden tranzystor na bit.
- Potrzebuje odświeżania ("wyciekający" ładunek).
- Wartość wyjściowa jest analogowa.

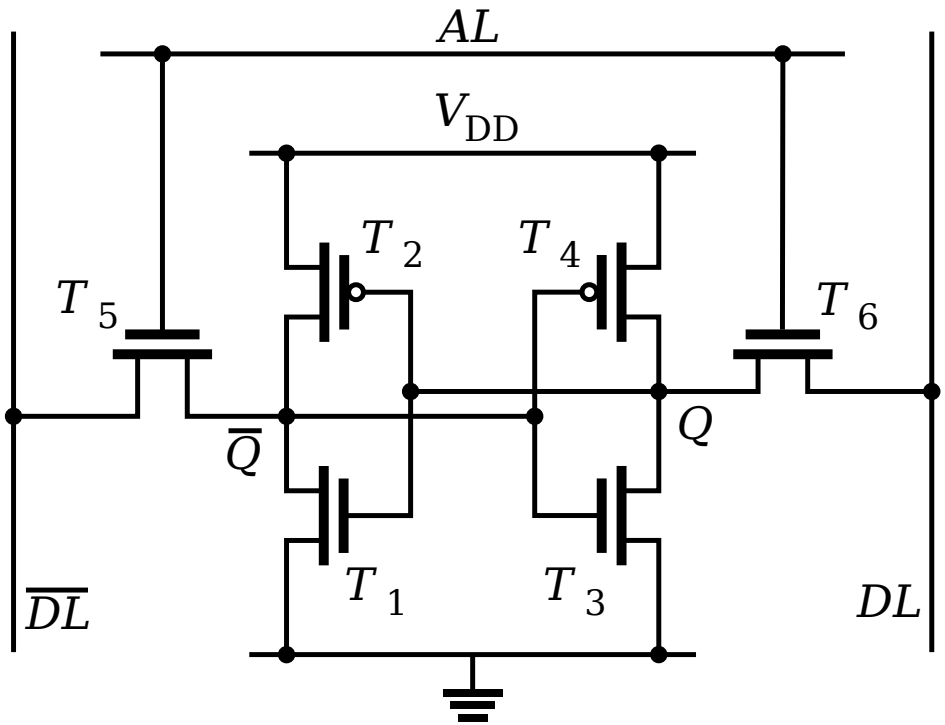
Cechy DRAM



Cechy DRAM:

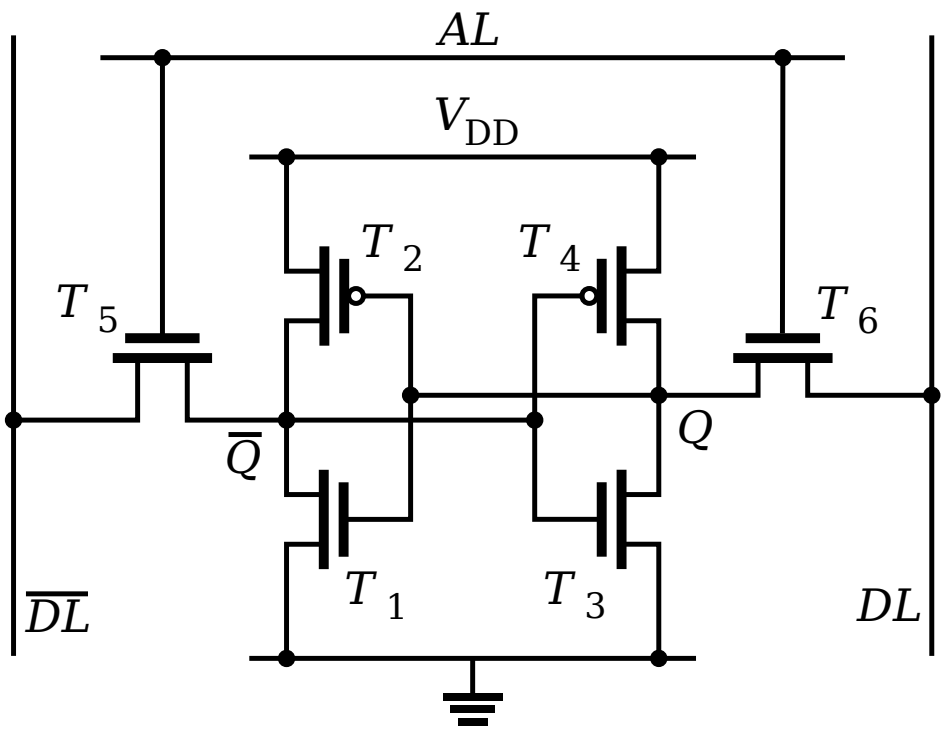
- Tylko jeden tranzystor na bit.
- Potrzebuje odświeżania ("wyciekający" ładunek).
- Wartość wyjściowa jest analogowa.
- Jest używany w zewnętrznych układach.

1 bit pamięci SRAM



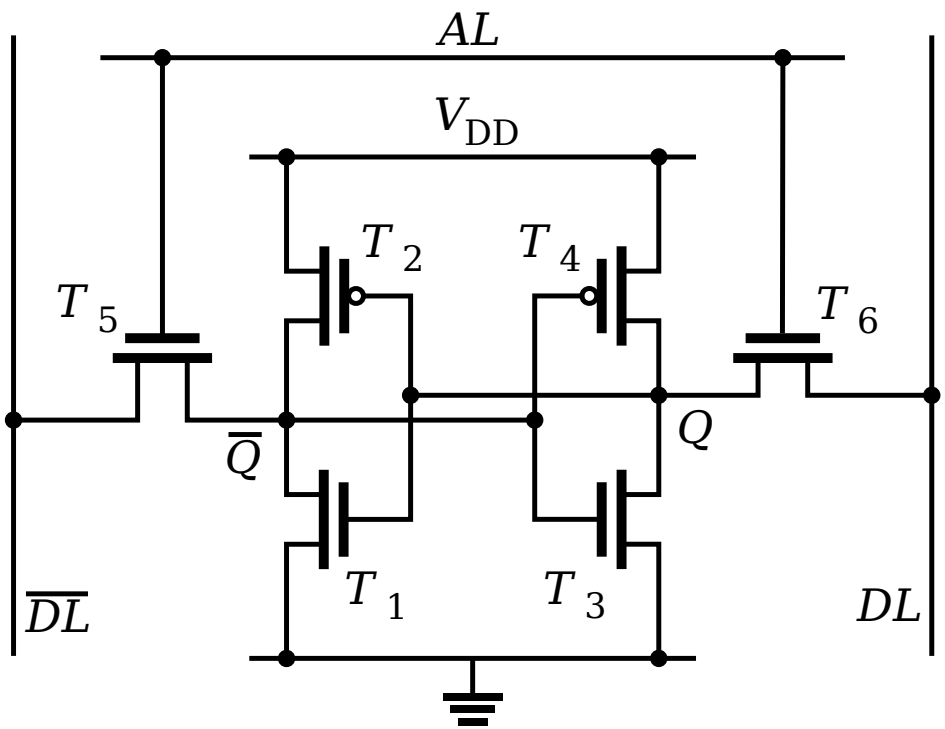
- AL - linia adresująca
- DL - linia danych
- T_{1-6} - tranzystory
- V_{DD} - Napięcie zasilające
- Q - Ładunek z przerzutnika

Cechy SRAM



Cechy SRAM:

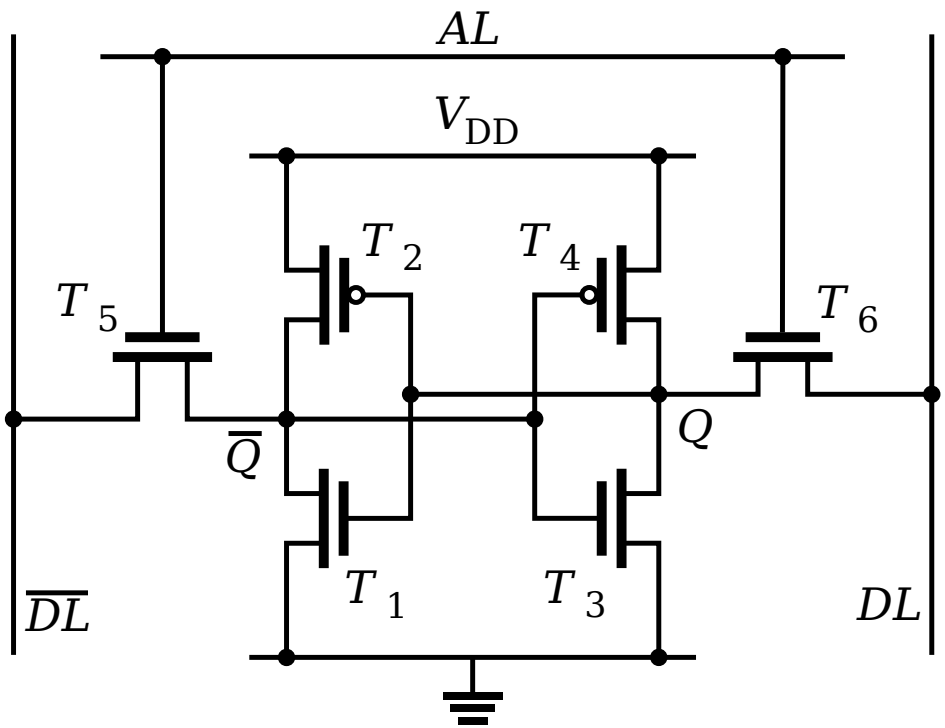
Cechy SRAM



Cechy SRAM:

- Aż 6 tranzystorów na każdy bit.

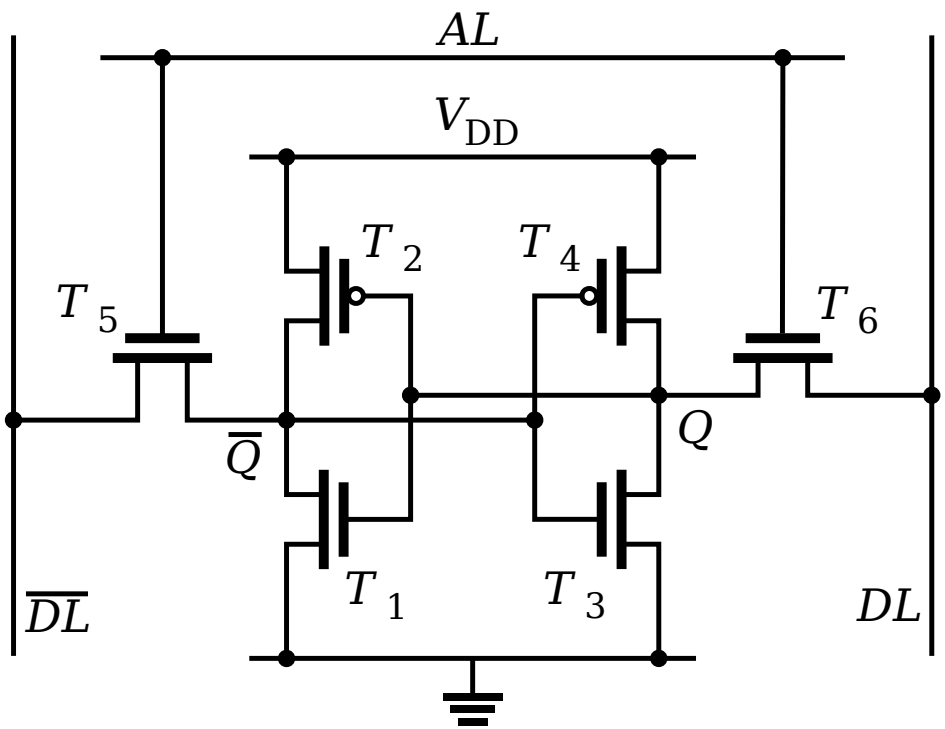
Cechy SRAM



Cechy SRAM:

- Aż 6 tranzystorów na każdy bit.
- Wymaga stałego zasilania V_{DD}.

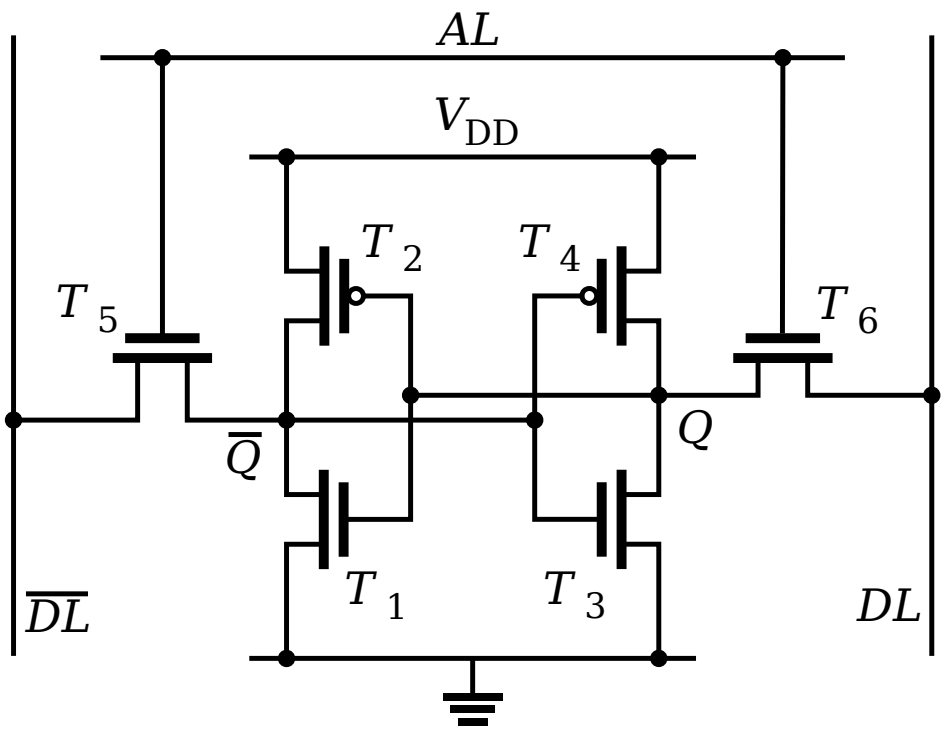
Cechy SRAM



Cechy SRAM:

- Aż 6 tranzystorów na każdy bit.
- Wymaga stałego zasilania V_{DD}.
- Wartości na wyjściach są cyfrowe.

Cechy SRAM



Cechy SRAM:

- Aż 6 tranzystorów na każdy bit.
- Wymaga stałego zasilania V_{DD}.
- Wartości na wyjściach są cyfrowe.
- Dostęp do wartości w czasie mniejszym niż 1ns.

Do czego nam pamięć podręczna?

Wiemy już następujące rzeczy:

- statyczny RAM jest drogi, mało pojemny ale bardzo szybki
- dynamiczny RAM jest tani, bardzo pojemny ale stosunkowo wolny.

Do czego nam pamięć podręczna?

Wiemy już następujące rzeczy:

- statyczny RAM jest drogi, mało pojemny ale bardzo szybki
- dynamiczny RAM jest tani, bardzo pojemny ale stosunkowo wolny.

Pytanie:

Jak zatem uzyskać dużą pojemności pamięci operacyjnej, zachowując przy tym wysoką prędkość?

Do czego nam pamięć podręczna?

Wiemy już następujące rzeczy:

- statyczny RAM jest drogi, mało pojemny ale bardzo szybki
- dynamiczny RAM jest tani, bardzo pojemny ale stosunkowo wolny.

Pytanie:

Jak zatem uzyskać dużą pojemności pamięci operacyjnej, zachowując przy tym wysoką prędkość?

Odpowiedź:

Poprzez wykorzystanie **pamięci podręcznej** (cache).

Czym jest pamięć podręczna?

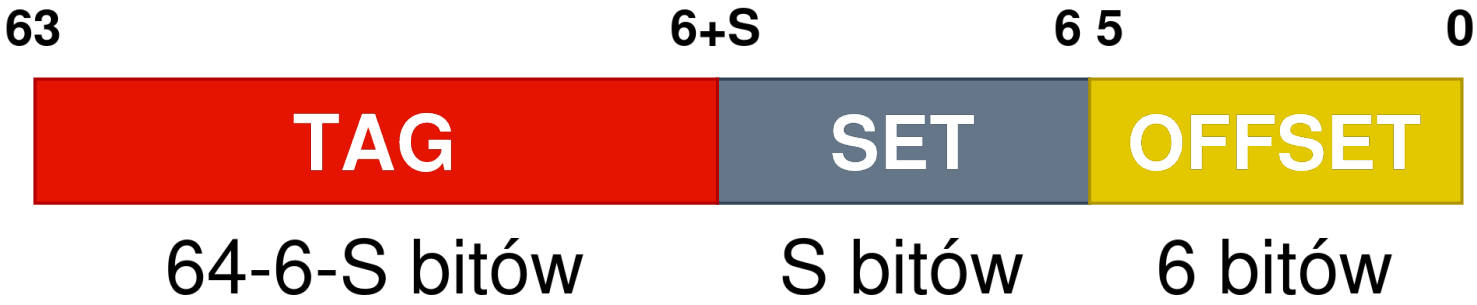
Cache

Cache to segment pamięci służącej do przechowywania kopii danych z pamięci o większej pojemności, do których dostęp będzie potrzebny w najlbiższej przyszłości.

Współczesne procesory często mają kilka poziomów pamięci podręcznej.

Adresowanie

Adresy w pamięci cache często są 64 bitowymi wartościami które są podzielone na 3 części jak na rysunku poniżej.



OFFSET to adresowanie poszczególnych bitów w wybranej poprzez SET oraz TAG linii pamięci podręcznej (ang. *cache line*). 6 bitów przeznaczonych jest dla nią dla tego że zwykle linia jest 64 bitowa.

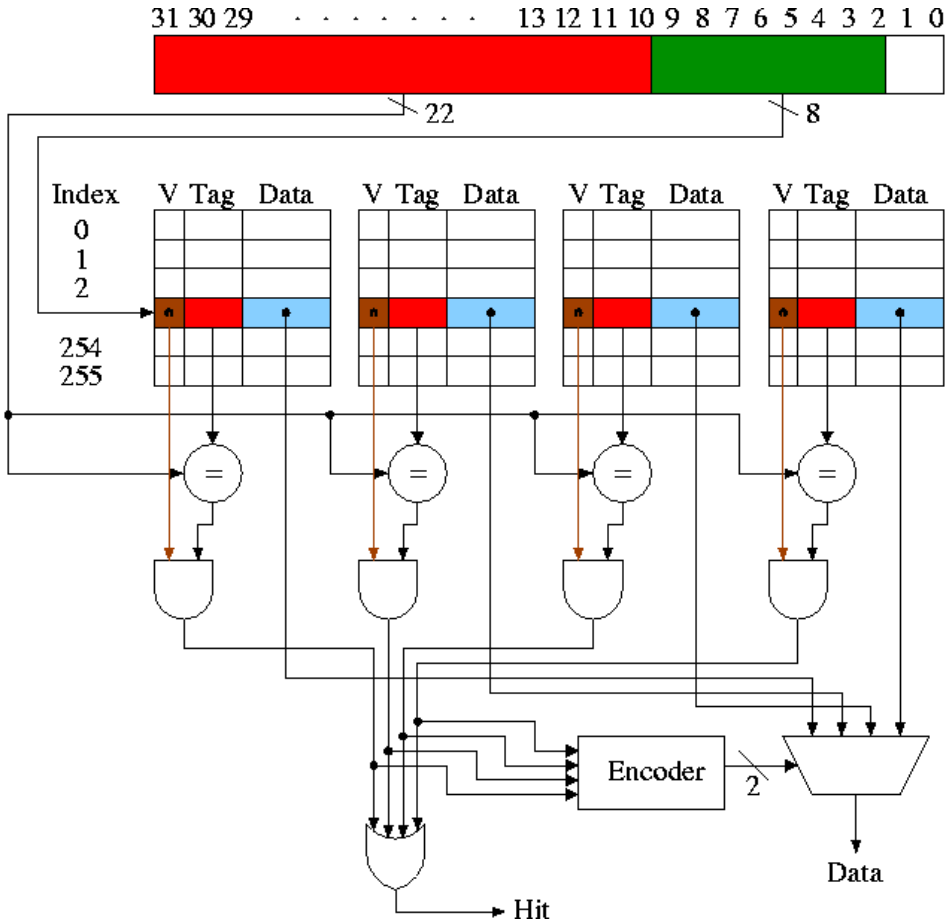
Adresowanie

Liczba S bitów przeznaczona na zbiór linii jest wyznaczana w zależności od tego ilo-"drożny" jest cache (ang. *n-way cache*).

Przykład

Weźmy pod uwagę przypadek gdzie linia ma 64 bity, pamięć jest 8-drożna i jej pojemność to 64KB. Chcemy znaleźć liczbę S bitów w sekcji SET adresu. Każdy zbiór zawiera w sobie 8 linii (dlatego, że pamięć 8-drożna) więc jeden zbiór zawiera $16 * 8 = 128$ bajtów. pojemność całego cache-u to $64KB$ więc istnieje $64KB/128B = 512$ zbiorów. Dla podanego przykładu $S = 9$ ponieważ $2^9 = 512$.

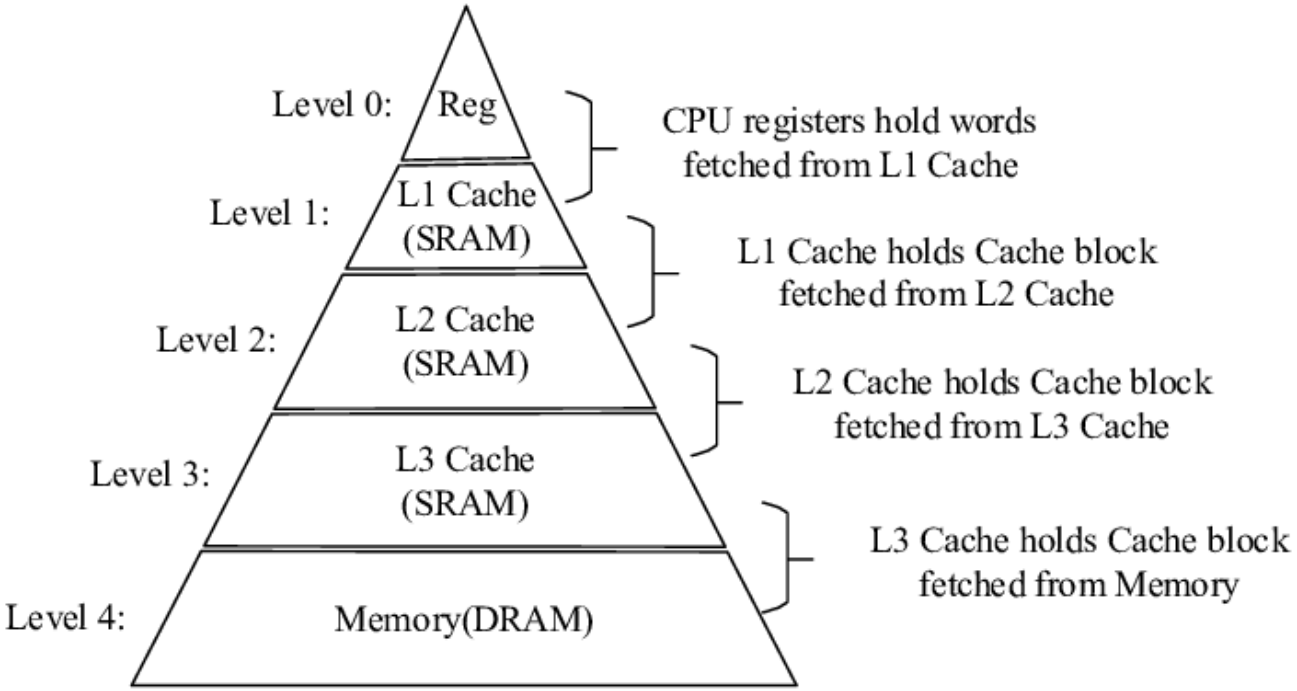
Adresowanie



Po co kilka poziomów?

Obecnie procesory mają zwykle 3 lub nawet 4 poziomy pamięci podręcznej. Dzieje się tak dlatego, że wraz ze zwiększaniem wielkości bloku pamięci, zwiększa się także opóźnienie związane z adresowaniem, a także rozmiar multiplekserów na kostce procesora. Poziomowanie rozwiązuje też do pewnego stopnia kwestię jednoczesnego dostępu do pamięci gdyż najniższe poziomy pamięci podręcznej (L0 - L1 - L2) są osobne dla każdego rdzenia, podczas gdy cache L3 jest współdzielony.

Hierarchia poziomów cache



Poziomy cd.

Przykładowo procesory z serii Ryzen 7000 mają kolejno

- L1 cache – 64 KB na rdzeń
- L2 cache – 1 MB na rdzeń
- L3 cache – 32 to 128 MB współdzielone

Poziomy cd.

Podczas próby dostępu do pamięci procesor w pierwszej kolejności szuka jej w najniższym poziomie pamięci podręcznej (zwykle nazwanej L0 lub L1). Jeśli nie znaleziono tych danych które były potrzebne, następuje **cache miss** i dane szukane są dalej w wyższych poziomach pamięci.

Równoległość a pamięć podręczna

Jakie problemy należy rozwiązać przy podzieleniu pamięci podręcznej na różne poziomy?

Równoległość a pamięć podręczna

Jakie problemy należy rozwiązać przy podzieleniu pamięci podręcznej na różne poziomy?

- Co się dzieje przy zapisie?

Równoległość a pamięć podręczna

Jakie problemy należy rozwiązać przy podzieleniu pamięci podręcznej na różne poziomy?

- Co się dzieje przy zapisie?
- Jak uzyskać spójny obraz pamięci podręcznej dla każdego rdzenia w procesorze?

MESI

Z pomocą przychodzi protokół MESI. Jego założeniem jest przypisywanie do każdej linii pamięci podręcznej jednego z czterech stanów:

MESI

Z pomocą przychodzi protokół MESI. Jego założeniem jest przypisywanie do każdej linii pamięci podręcznej jednego z czterech stanów:

- M (modified) - linia pamięci jest dostępna tylko w jednym z poziomów cache i jest różna od zawartości pamięci głównej.

MESI

Z pomocą przychodzi protokół MESI. Jego założeniem jest przypisywanie do każdej linii pamięci podręcznej jednego z czterech stanów:

- M (modified) - linia pamięci jest dostępna tylko w jednym z poziomów cache i jest różna od zawartości pamięci głównej.
- E (exclusive) – linia pamięci jest dostępna tylko w jednym z poziomów cache, oraz w pamięci operacyjnej

MESI

Z pomocą przychodzi protokół MESI. Jego założeniem jest przypisywanie do każdej linii pamięci podręcznej jednego z czterech stanów:

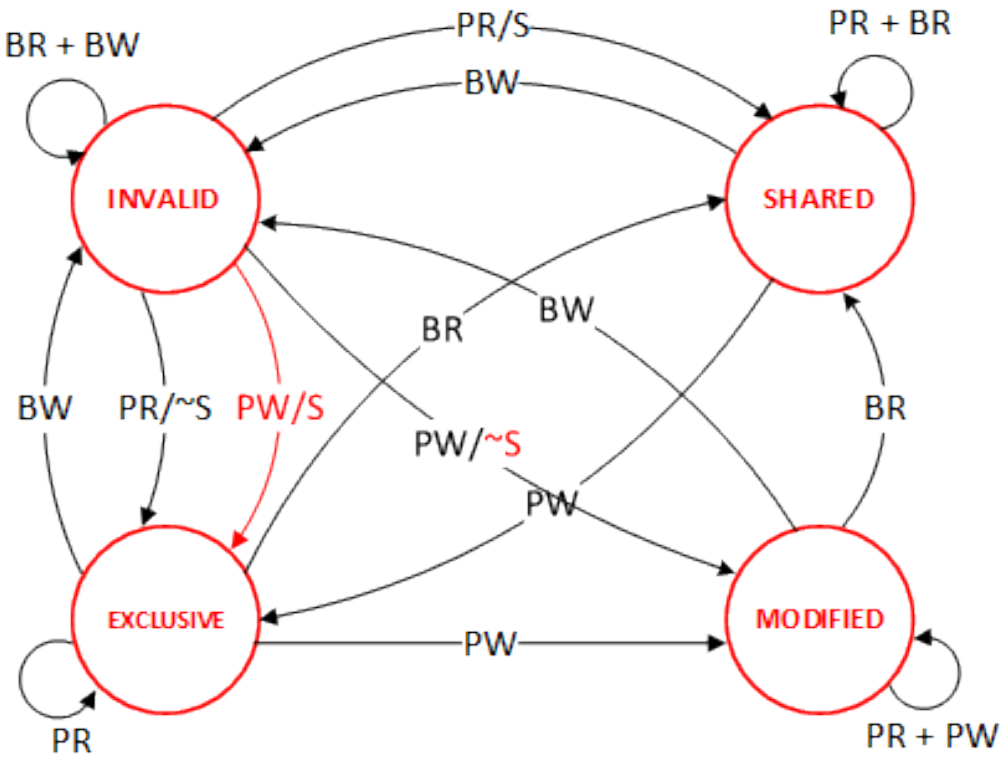
- M (modified) - linia pamięci jest dostępna tylko w jednym z poziomów cache i jest różna od zawartości pamięci głównej.
- E (exclusive) – linia pamięci jest dostępna tylko w jednym z poziomów cache, oraz w pamięci operacyjnej
- S (shared) – linia pamięci jest dostępna na wszystkich poziomach cache oraz w pamięci operacyjnej

MESI

Z pomocą przychodzi protokół MESI. Jego założeniem jest przypisywanie do każdej linii pamięci podręcznej jednego z czterech stanów:

- M (modified) - linia pamięci jest dostępna tylko w jednym z poziomów cache i jest różna od zawartości pamięci głównej.
- E (exclusive) – linia pamięci jest dostępna tylko w jednym z poziomów cache, oraz w pamięci operacyjnej
- S (shared) – linia pamięci jest dostępna na wszystkich poziomach cache oraz w pamięci operacyjnej
- I (invalid) – linia pamięci jest nieaktualna i może zostać zastąpiona inną.

Diagram przejść w MESI



PR	= processor read	BR	= observed bus read
PW	= processor write	BW	= observed bus read
S/~S	= shared/NOT shared		

Co robić?

- Dobre wykorzystywanie pamięci podręcznej

Co robić?

- Dobre wykorzystywanie pamięci podręcznej
  - utrzymywać małe bloki danych.

Co robić?

- Dobre wykorzystywanie pamięci podręcznej
  - utrzymywać małe bloki danych.
  - próbować skanować liniowo.

Co robić?

- Dobre wykorzystywanie pamięci podręcznej
  - utrzymywać małe bloki danych.
  - próbować skanować liniowo.
  - unikać skakania po wskaźnikach.

Co robić?

- Dobre wykorzystywanie pamięci podręcznej
  - utrzymywać małe bloki danych.
  - próbować skanować liniowo.
  - unikać skakania po wskaźnikach.
- Prefetching

Co robić?

- Dobre wykorzystywanie pamięci podręcznej
  - utrzymywać małe bloki danych.
  - próbować skanować liniowo.
  - unikać skakania po wskaźnikach.
- Prefetching
  - Prefetching to pobieranie danych w dużych kawałkach przewidując ich wykorzystanie w bliskim czasie.

Co robić?

- Dobre wykorzystywanie pamięci podręcznej
  - utrzymywać małe bloki danych.
  - próbować skanować liniowo.
  - unikać skakania po wskaźnikach.
- Prefetching
  - Prefetching to pobieranie danych w dużych kawałkach przewidując ich wykorzystanie w bliskim czasie.
  - umożliwiać procesorowi automatyczny prefetching

Co robić?

- Dobre wykorzystywanie pamięci podręcznej
  - utrzymywać małe bloki danych.
  - próbować skanować liniowo.
  - unikać skakania po wskaźnikach.
- Prefetching
  - Prefetching to pobieranie danych w dużych kawałkach przewidując ich wykorzystanie w bliskim czasie.
  - umożliwiać procesorowi automatyczny prefetching
  - dokonywać tego samodzielnie (trudniejsze zagadnienie)

Źródła

- https://www.youtube.com/watch?v=4_smHyqgDTU
- https://www.akkadia.org/drepper/cpumemory.pdf
- https://cs.nyu.edu/~gottlieb/courses/2000s/2001-02-fall/arch/lectures/lecture-22.html
- https://pl.wikipedia.org/wiki/MESI