

Systemy komputerowe: architektura i programowanie (SYKOM)

Wykład: Pamięci

Aleksander Pruszkowski

Instytut Telekomunikacji Politechniki Warszawskiej

PLAN WYKŁADY

- Pamięci trwałe – ROM/PROM/EPROM/FLASH ROM
- Pamięci ulotne – Static RAM
- Pamięci ulotne – Dynamic RAM
- Tryby pracy i odświeżanie pamięci dynamicznych
- Hierarchia pamięci
- Łączenie pamięci
- Dekoder pamięci
- Budowa pamięci podręcznych
- Detekcja i korekcja błędów w pamięciach

Pamięci trwałe - ROM/PROM/EPROM/FLASH ROM

- Budowa pamięci - ROM (zwana też „Mask ROM”)
 - Pamięć programowana w procesie projektowania masek używanych w procesie produkcji pamięci
 - brak możliwości kasowania i programowania w urządzeniu
- Budowa pamięci - PROM
 - Komórka pamięciowa zawiera „bezpiecznik” przepalany podczas programowania
 - możliwe jednorazowe zaprogramowanie treści, z marginalną możliwością modyfikacji treści (przez późniejsze przepalenie nie „tkniętych” bezpieczników)
- Budowa pamięci - EPROM
 - Chwilowe przebicie bramki tranzystora MOS umożliwia jej naładowanie ładunkiem który tam jest odtąd uwięziony i wyznacza stan komórki - wymagane wyższe napięcie ($\sim 21V$)
 - Odprowadzenie ładunków z wszystkich komórek(!) możliwe wyłącznie poprzez naświetlenie przez specjalne okienko promieniami UV

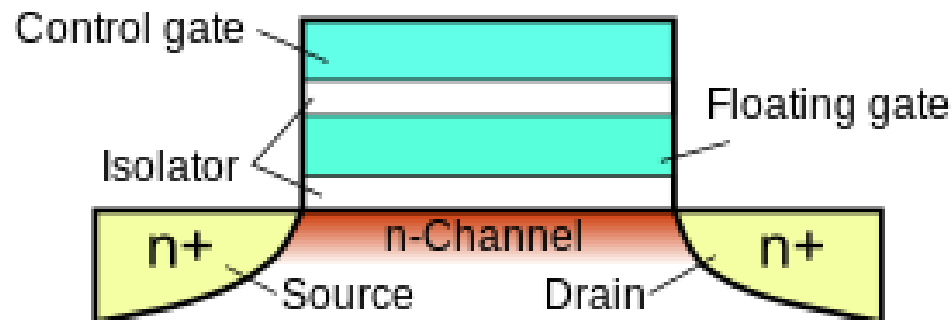
Pamięci trwałe - ROM/PROM/EPROM/FLASH ROM

■ Budowa pamięci EEPROM

- Pływająca bramka może uwięzić ładunek - podobnie jak w EPROM
- Ładunek ten można usunąć za pomocą odpowiednio przyłożonego napięcia

■ Tryby pracy/szybkości

- Podstawa pracy: kasowanie (zmiana zawartości na „1”) i wpisywanie do komórek żądanych wartości („1” pozostaje nie tknięte, wpisuje się tylko „0”)
- Kasowanie
 - typowo pamięć kasowana jest w całości, może być „wbudowane” w operacje zapisu
 - procedura czasochłonna - specjalny „pin” (np.: IO7) lub specjalny rejestr (może być czytany w dowolnym momencie) wskazuje koniec operacji



Pamięci trwałe - ROM/PROM/EPROM/FLASH ROM

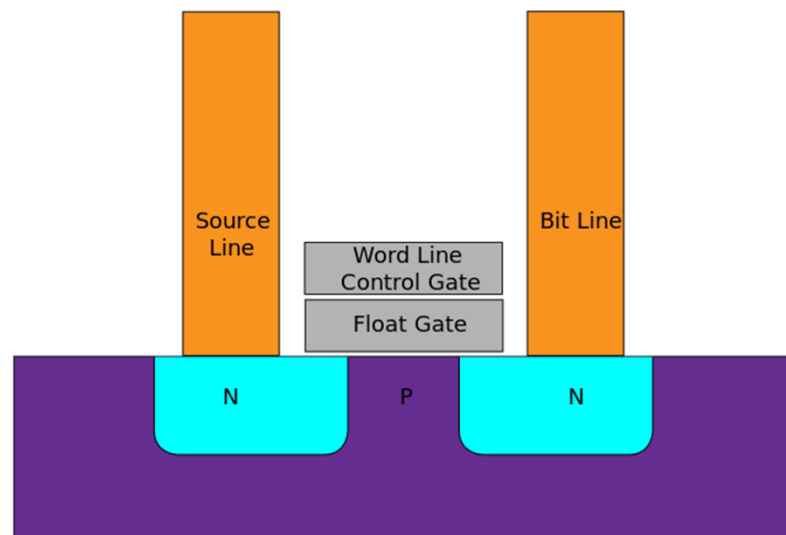
■ Parametry przykładowych modułów

- Pojemność: 64B...64KB(CAT24C512), 1024B(ATMega328P), 128KB(AT28C010), 256KB(AT24CM02 - pamięć o największej pojemności)
- Energochłonność operacji
 - CAT24C512: 1mA - odczyt, 2,5mA - zapis, 2uA - spoczynek
 - AT28C010: 40mA - producent nie specyfikuje typu operacji, 200uA - spoczynek
 - AT24CM02: 2mA - odczyt, 3mA - zapis, 6uA - spoczynek
- Trwałość przy kasowaniu
 - ATMega328P: 10^5 cykli
 - CAT24C512: 10^6 cykli
 - AT28C010: $10^4..10^5$ cykli
 - AT24CM02: 10^6 cykli

Pamięci trwałe - ROM/PROM/EPROM/FLASH ROM

■ Budowa pamięci

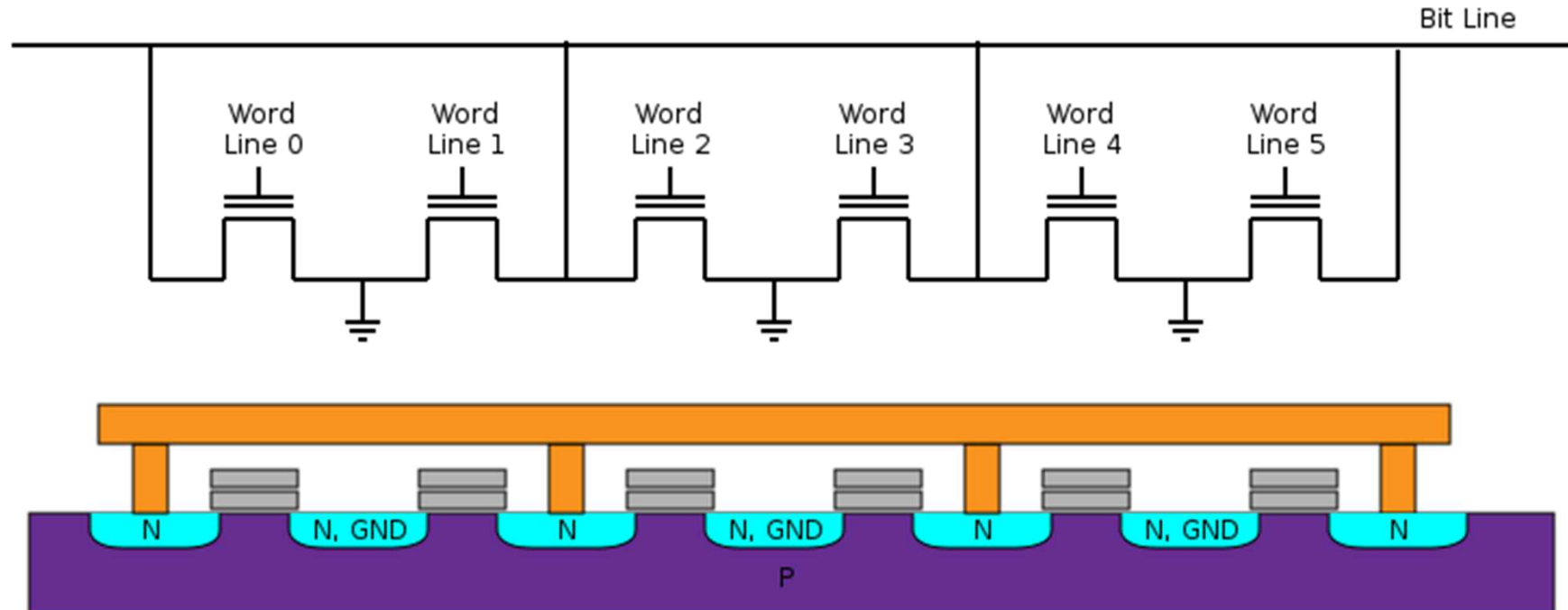
- Komórka pamięci zbudowana w oparciu o tranzystor MOS z pływającą bramką
- Technologia wywodzi się z pamięci EEPROM i EPROM
- Zawierają wbudowane przetwornice podwyższające napięcie
 - brak elementów zewnętrznych jest okupione małą jej efektywnością energetyczną
- Ze względu na budowę istnieją dwa typy: NOR i NAND



Pamięci trwałe - ROM/PROM/EPROM/FLASH ROM

■ Budowa pamięci - NOR

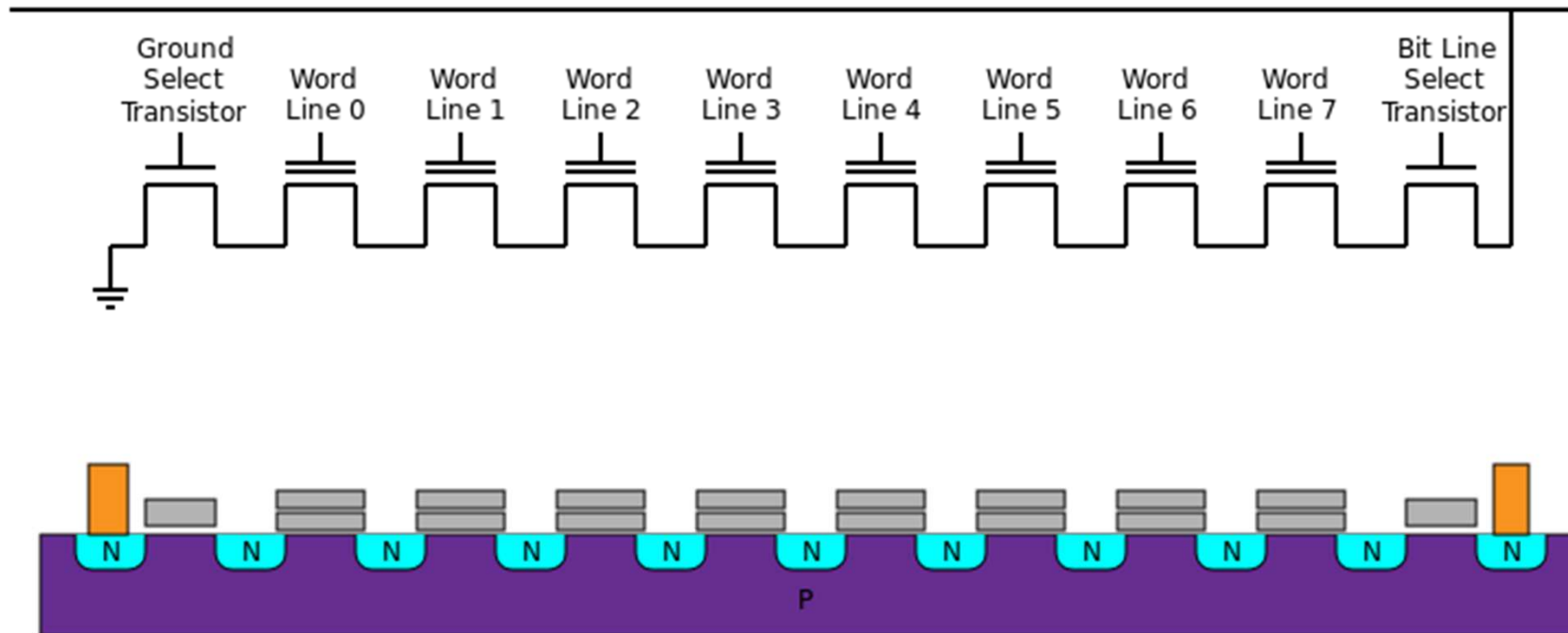
- Bezpośredni dostęp do każdej komórki pamięci
- Szybka operacja czytania, długie czasy zapisu i kasowania
- Dobrze sprawdza się jako pamięć „firmware” (sporadycznie modyfikowanie)
- Trwałość: $10^4..10^5$ cykli kasowania
- Mała gęstość komórek



Pamięci trwałe - ROM/PROM/EPROM/FLASH ROM

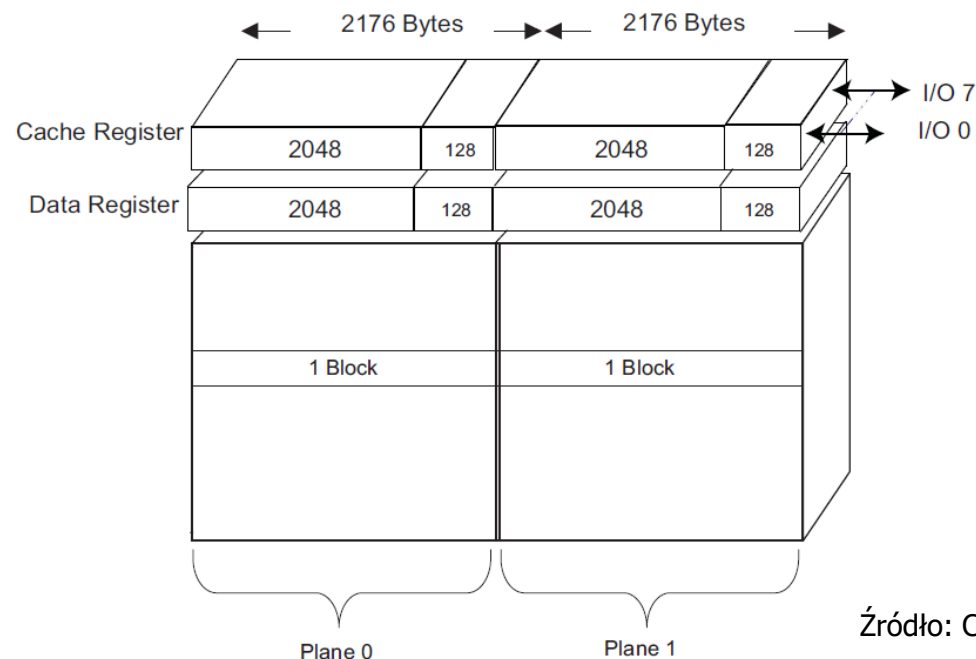
■ Budowa pamięci - NAND

- Sekwencyjny dostęp do komórek pamięci (cała strona)
- Wolniejszy odczyt, szybsze czasy zapisu i kasowania
- Dobrze sprawdza się jako pamięć masowa
- Trwałość: $<10^6$ cykli kasowania (tylko SLC)
- Duża gęstość komórek



Pamięci trwałe - ROM/PROM/EPROM/FLASH ROM

- Parametry przykładowych modułów - S34ML04G2
 - napięcie pracy: 2,7...3,3V
 - pobory prądu:
 - faza włączania: 50mA
 - czytanie/pisanie/kasowanie: <30mA
 - uśpienie: 1mA(/CE zgodne z stanem „1” TTL), 50uA(gdy /CE bliskie VCC)
 - organizacja: 2x plane, plane: 2048 bloków, blok: 64 strony, strona: 2176B (treść: 2048B + spare: 128B)



Pamięci trwałe - ROM/PROM/EPROM/FLASH ROM

- Parametry przykładowych modułów - S34ML04G2, cd.
 - trwałość: 10^6
 - czytanie strony: 25us
 - programowanie strony: 300us
 - kasowanie bloku: 3ms
 - operacje równoległe dla każdej części (plane) - zwiększają wydajność
- W projektach IoT warte rozważenia są pamięci micro-SD: będące hybrydą pamięci i mikrokontrolera jej obsługujące
 - zainstalowane w tych MCU są specjalne komponenty (wielokanałowe DMA, obliczanie CRC/ECC, ...) zwiększające wydajność operacji

Pamięci trwałe - ROM/PROM/EPROM/FLASH ROM

■ Współpraca z zewnętrznymi pamięciami FLASH, cd.

■ Bloki i strony w typowych pamięciach FLASH

- podstawowa jednostka kasowania - blok/sektor
- podstawowa jednostka zapisu - strona

Przykłady z podziału na bloki/sektory i strony

■ ST25P80

- pojemność 1MB, blok/sektor 128KB, strona 256B
 - jeden blok to 1024 strony, cała pamięć to tylko 8 bloków(!)

■ AT45DB041

- pojemność 512KB, blok/sektor 256B, strona 256B
 - jeden blok to jedna strona(!), cała pamięć to 2048 bloków/sektorów

■ GD25Q128C

- pojemność 128MB, blok/sektor 64KB, strona 256B
 - jeden blok to 256 stron, cała pamięć to tylko 2048 bloków

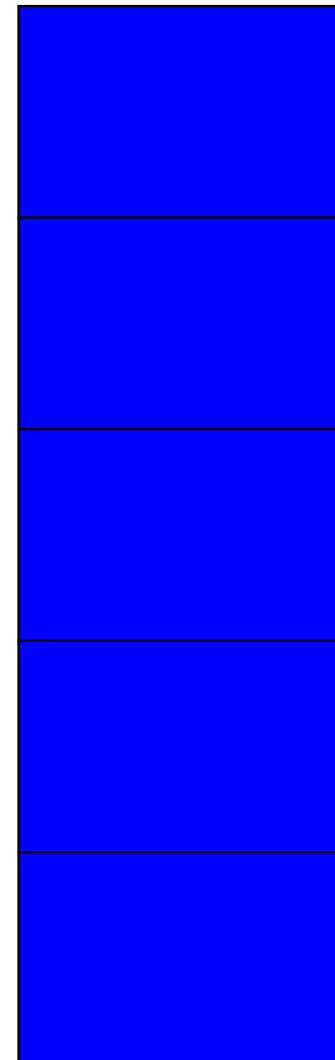
■ GD5F2GQ4UCFIG

- pojemność 2GB, blok/sektor 128K+8KB, strona 2KB+128B
 - jeden blok to 64 strony, cała pamięć to tylko 16384 bloków

■ W25Q32

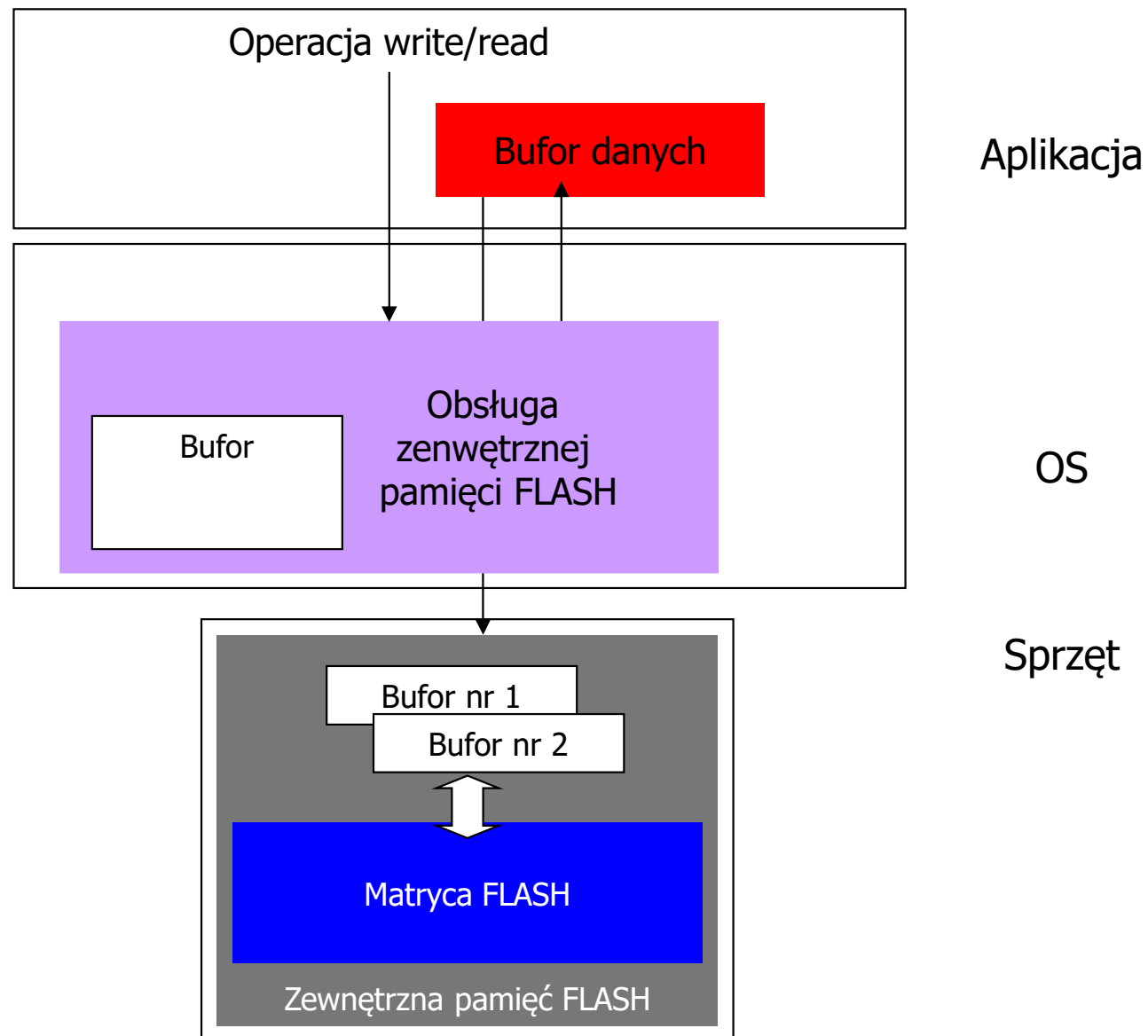
- pojemność 4GB, blok 64KB, sektor 4KB, strona 256B
 - jeden blok to 256 stron, cała pamięć to tylko 65536 bloków
 - w tej pamięci minimalna jednostka kasowania to sektor(!)
 - czas kasowania sektora to <200ms, gdy bloku <1,5s

Pamięć Flash



Pamięci trwałe - ROM/PROM/EPROM/FLASH ROM

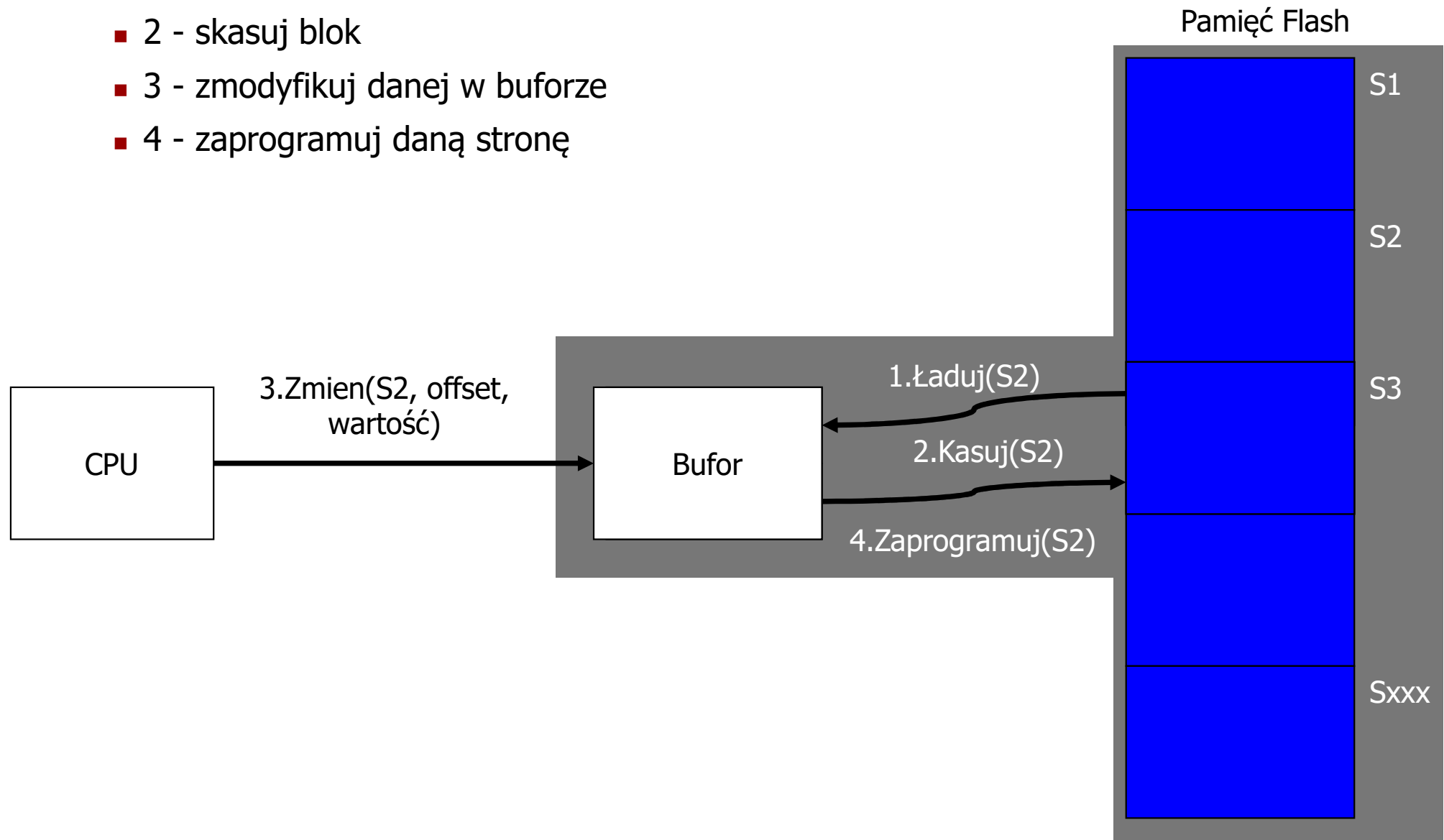
■ Współpraca z zewnętrznymi pamięciami FLASH



Pamięci trwałe - ROM/PROM/EPROM/FLASH ROM

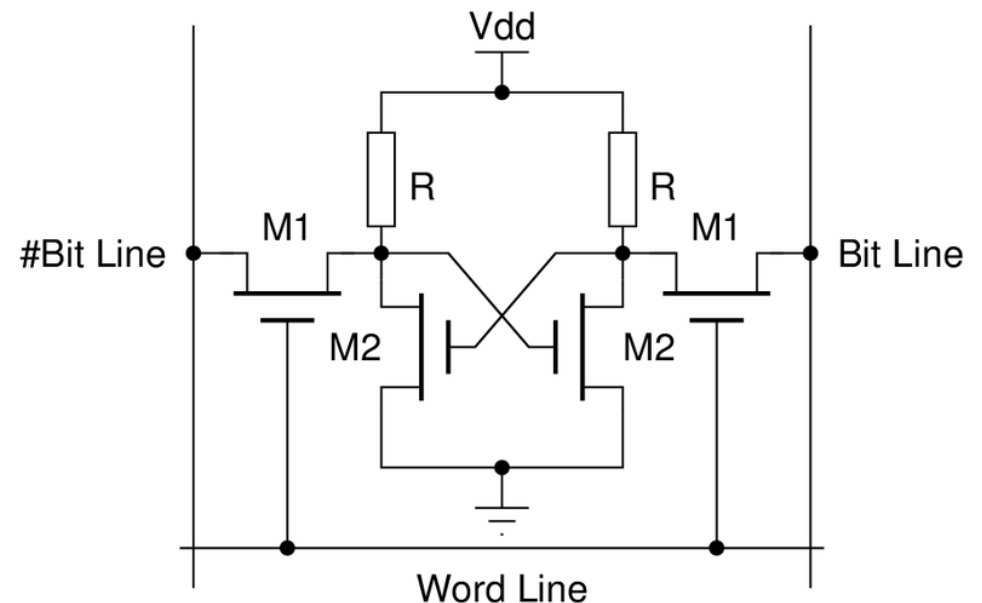
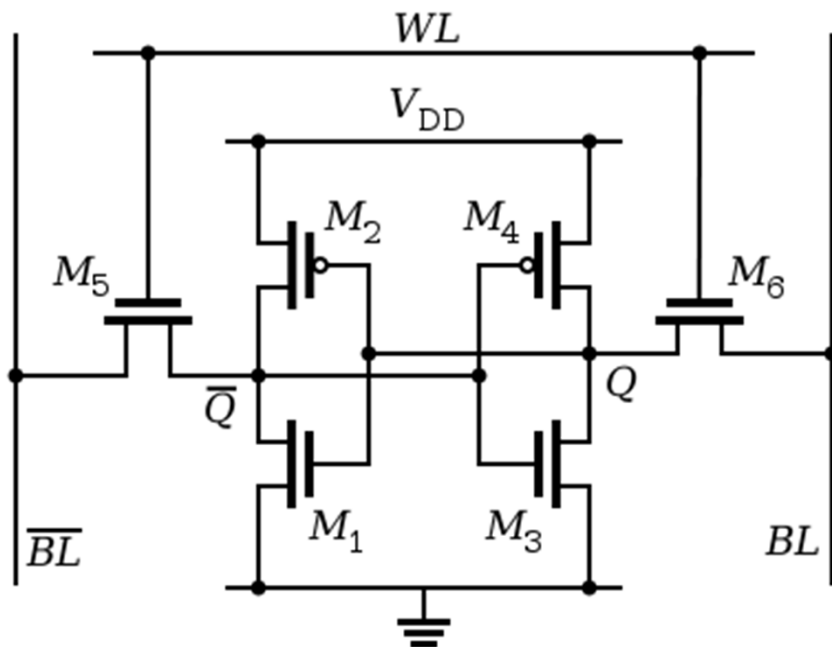
■ Bloki i strony w typowych pamięciach FLASH

- Operacja modyfikacji danych (dostęp swobodny)
 - 1 - pobierz do bufora dane z danej strony
 - 2 - skasuj blok
 - 3 - zmodyfikuj danej w buforze
 - 4 - zaprogramuj daną stronę



Pamięci ulotne – Static RAM

- Budowa pamięci statycznych SRAM (ang. Static RAM)
 - element pamięciowy to przerzutnik
 - typowo zbudowany z 6/4 tranzystorów
 - treść po włączeniu zasilania nie określona
 - wysoka szybkość operacji zapisu/odczytu



Pamięci ulotne – Static RAM

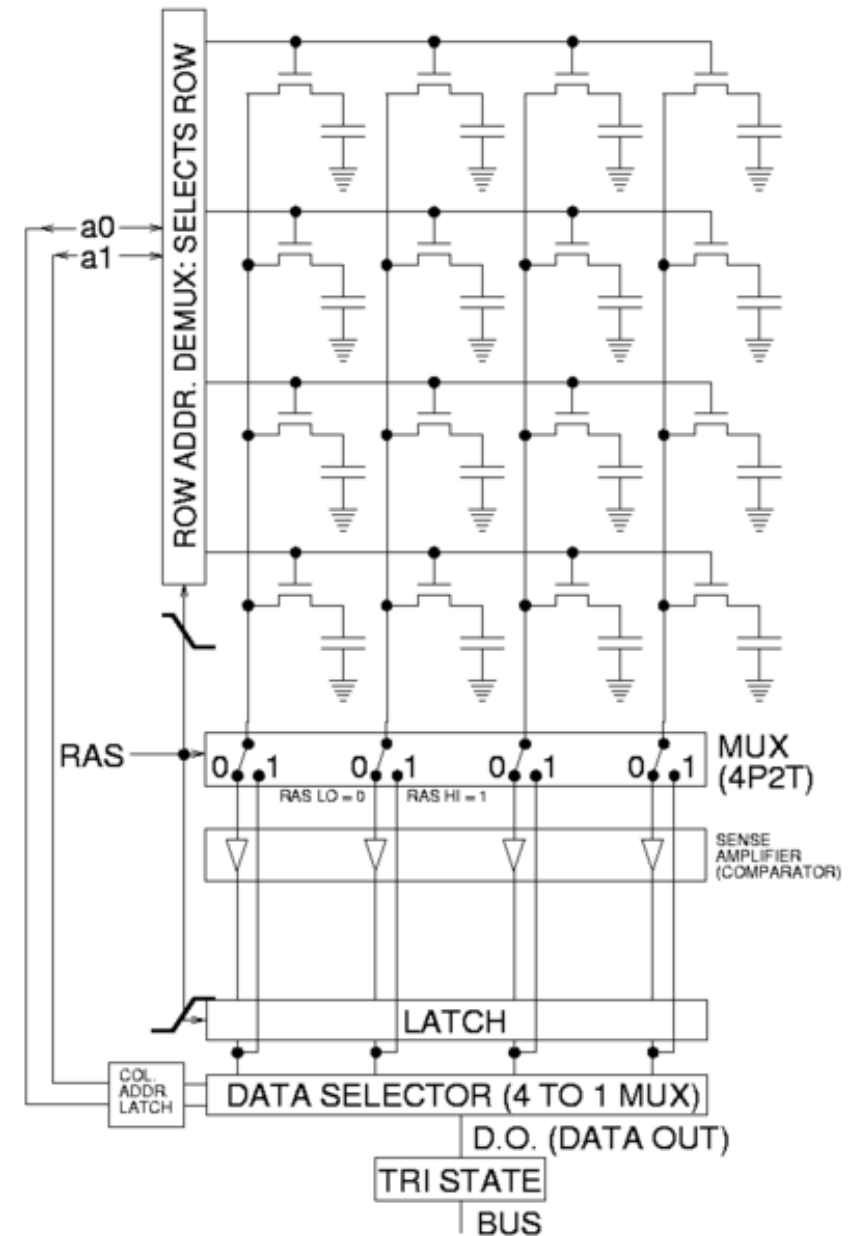
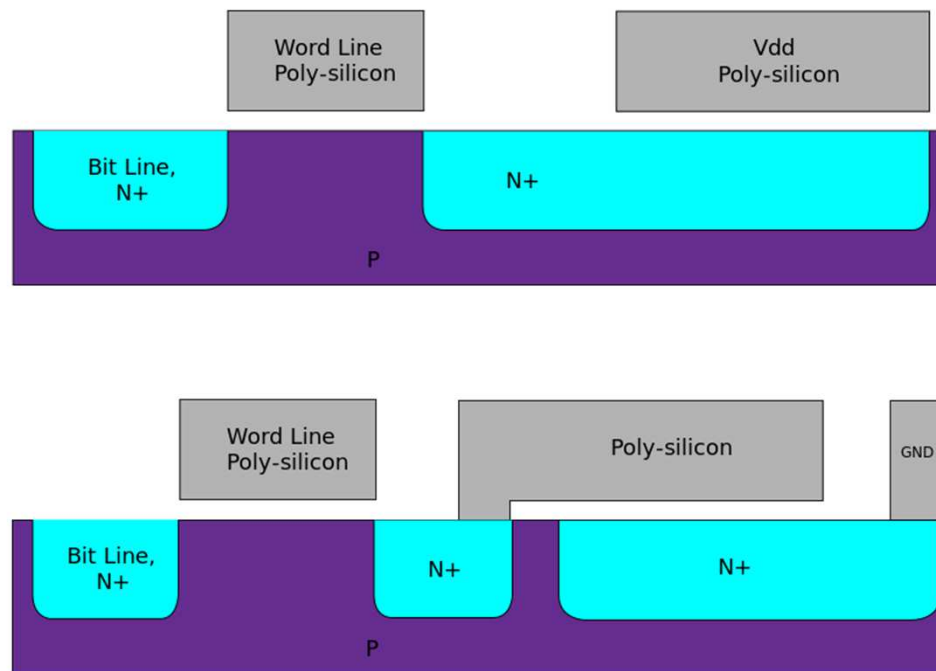
■ Parametry przykładowych modułów SRAM

- Obecnie występują jako pojedyncze moduły w wersjach Dual-port - GS82582Q18GE
 - napięcie pracy: 1,7...1,9V
 - 16M słów x 18bitów
 - czasy dostępu <4ns (dostęp sekwencyjny)
 - pobory prądów zapis/odczyt: ~1A(!), uśpienie: 0,5A(!)
- Nieliczne moduły można podłączać via SPI/I2C - 23LC1024T
 - napięcie pracy: 2,5...5V, 128KB (32B strony)
 - czasy dostępu - wyznacza transfer SPI
 - pobory prądów zapis/odczyt: 10mA, uśpienie: <10uA (utrzymanie danych możliwe przy zasilaniu >1V)
- SRAM wbudowane w MCU – czasy dostępu na poziomie czasu przetwarzania instrukcji
 - ATmega328p: 2KB - przechowywanie w uśpieniu: już od 0,1uA
 - MSP430Fxxx: od 256B...10KB - przechowywanie w uśpieniu: już od 0,1uA

Pamięci ulotne – Dynamic RAM

■ Budowa pamięci dynamicznych DRAM (ang. Dynamic RAM)

- Przechowywanie danych - w pojemności sterowanej tranzystorem MOS
- Problem upływności pojemności obchodzone cyklicznym odświeżaniem



Tryby pracy i odświeżanie pamięci dynamicznych

Tryby pracy i odświeżanie pamięci dynamicznych

■ Pamięci DRAM typy

■ Asynchroniczne - bez synchronizacji z magistralą systemową

■ FPM DRAM (Fast Page Mode DRAM)

- wersja podstawowa, szybsze odwołania w jednym wierszu (strony) - bez konieczności adresowania numeru wiersza podczas czytania kolejnych lokacji w tym wierszu

■ EDO RAM (Extended Data Output RAM)

- wersja ulepszona o możliwość rozpoczęcia nowego cyklu podczas kończenia poprzedniego – czytamy/piszemy do obecnej lokacji ustawiając równocześnie nową lokację

■ BEDO DRAM (Burst EDO DRAM)

- wersja pozwalająca na seryjny transfer danych - wbudowany w pamięć licznik adresujący kolejne lokacje w wierszu

Tryby pracy i odświeżanie pamięci dynamicznych

■ Pamięci DRAM typy, cd.

■ Synchroniczne DRAM (SDRAM) – zsynchronizowane z magistralą systemową

■ Wprowadzały mechanizmy

■ praca z przeplotem

- możliwość równoczesnego pisania do banku gdy czytamy z innego banku
- wprowadzono także pojęcie „Rank” – określa liczbę zbiorów bloków mogących działać wspólnie

■ transfer pakietowy (burst)

- po ustaleniu adresu możliwość dostępu do sąsiednich lokacji – wsparcie dla pamięci podręcznych ładujących całe bloki

■ obsługa równoczesna wielu banków (typowo do czterech)

■ Tryby pracy

■ SDR (Single Data Rate)

- dane przesyłane przy każdym narastającym zboczach sygnału zegarowego

■ DDR (Double Data Rate)

- dane przesyłane przy narastającym i opadającym zboczach sygnału zegarowego (zwiększenie przepustowości)

■ DDR2, DDR3, DDR4, ... – zwiększają transfer przesyłając x2, x4, x8 więcej informacji w jednym takcie zegara

Tryby pracy i odświeżanie pamięci dynamicznych

■ Cykl odczytu pamięci dynamicznej

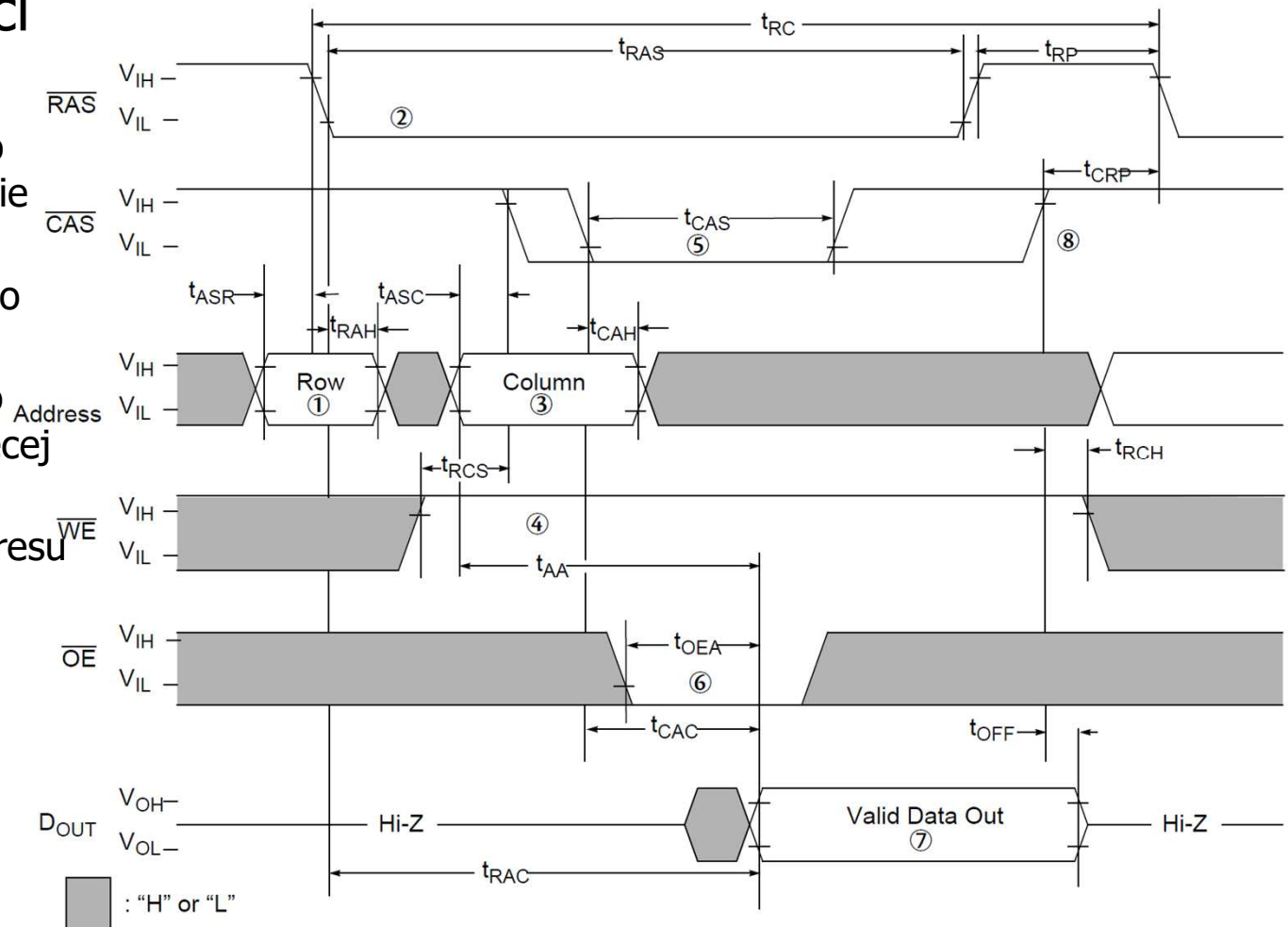
■ Definicje

- T_A – czas dostępu: od wystawienia adresu do pojawienia się danych
- T_C – czas cyklu: od wystawienia adresu do następnego cyklu adresowego
 - T_A i T_C mogą być różne
 - T_A może być większy niż T_C dla dostępu do kolejnych lokacji – bo pamięć wymaga zdekodowania pełnego adresu (rejstry CAS i RAS) a przy następnych odwołaniach wewnętrznych, tylko automat wyznacza adres (np.: +1)
- R_T – szybkość transferu ($1/T_C$)
- Transfer N jednostek informacji: $T_A + (N - 1) * T_C$

Tryby pracy i odświeżanie pamięci dynamicznych

■ Cykl odczytu z pamięci dynamicznej

- T_{ASR} – od adresu wiersza do sygnału /RAS (min – czyli nie mniej niż ...)
- T_{ASC} – od adresu kolumny do /CAS (min)
- T_{RAC} – do adresu wiersza do danych (max – czyli nie więcej niż ...)
- T_{AA} – od ostatniej części adresu do danych (max)
- T_{RC} – czas cyklu (max)
- ...

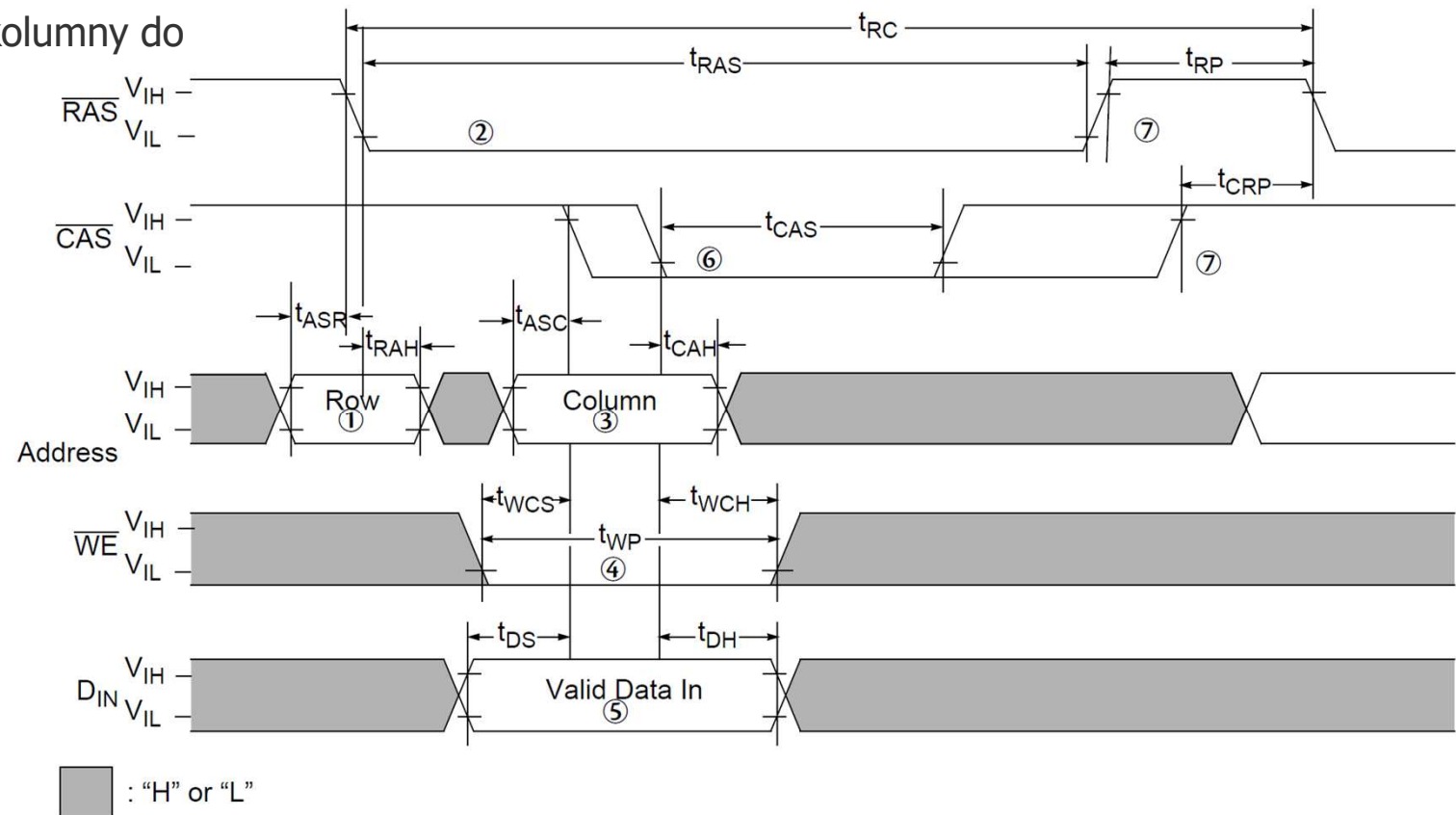


1 – wystawienie adresu wiersza (Address), 2 – zatrzaśnięcie wiersza (/RAS), 3 – wystawienie adresu kolumny (Address), 4 – ustalenie typu operacji (/WE), 5 – zatrzaśnięcie adresu kolumny (/CAS), 6 – uaktywnienie wyjścia pamięci (/OE), 7 – wyprowadzenie treści na magistralę danych (Dout), 8 – zdjęcie sygnałów (/CAS, /RAS, Dout=Hi-Z)

Tryby pracy i odświeżanie pamięci dynamicznych

■ Cykl zapisu do pamięci dynamicznej

- T_{ASR} – od adresu wiersza do sygnału $/RAS$
- T_{ASC} – od adresu kolumny do $/CAS$
- T_{RC} – czas cyklu



1 – wystawienie adresu wiersza (Address), 2 – zatrzaśnięcie wiersza ($/RAS$), 3 – wystawienie adresu kolumny (Address), 4 – ustalenie typu operacji ($/WE$), 5 – pojawienie się danych do zapisania do pamięci (D_{in}), 6 – zatrzaśnięcie adresu kolumny ($/CAS$), 7 – zdjęcie sygnałów ($/CAS$, $/RAS$)

■ Cykl zapisu do pamięci dynamicznej tryb EDO

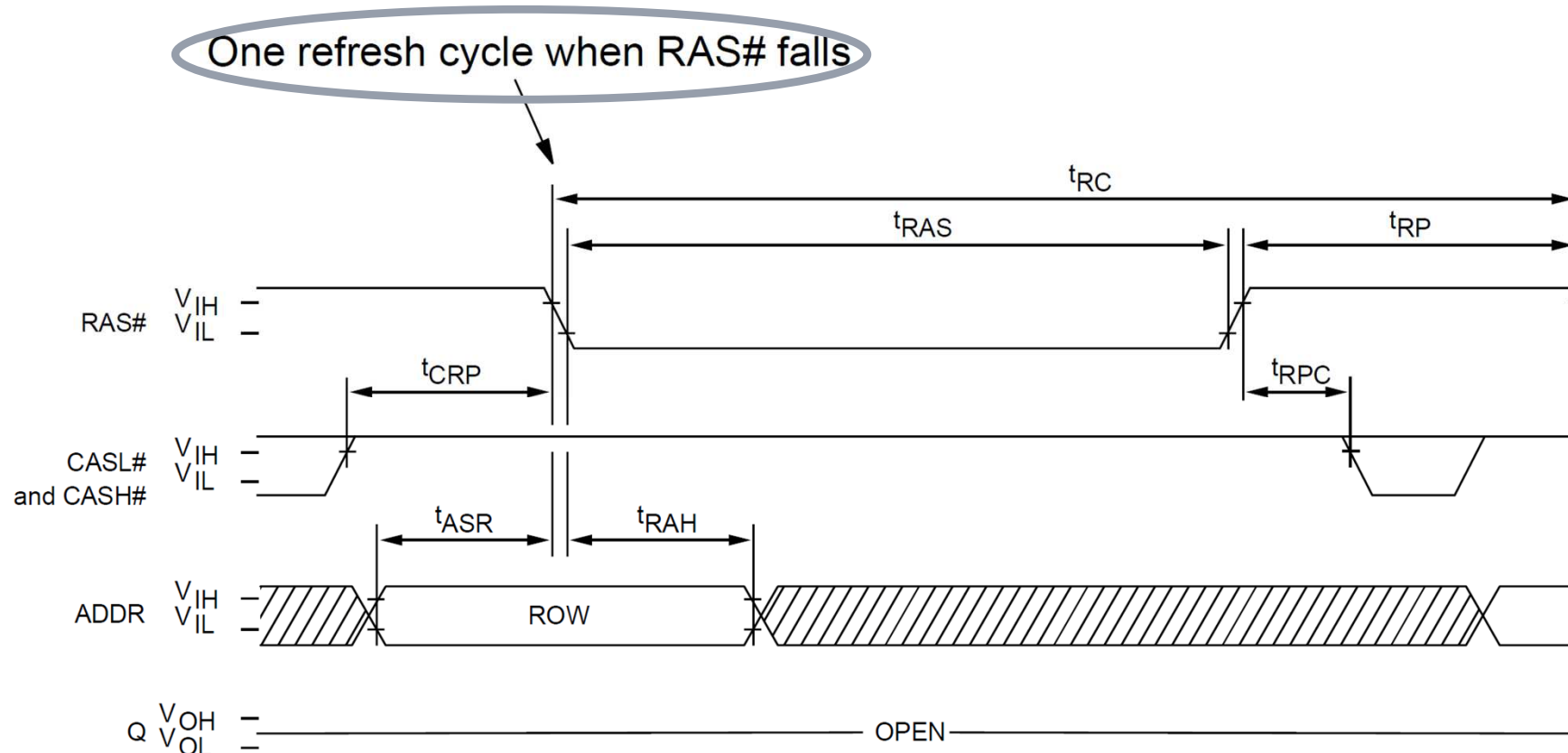


Tryby pracy i odświeżanie pamięci dynamicznych

■ Odświeżanie pamięci dynamiczne

■ Tryb wierszowy (ang. RAS only)

- 1) na magistrali adresowej podaje się adres wiersza od odświeżenia i sygnał /RAS
- 2) moduł pamięci identyfikuje to jako rozkaz rozpoczęcia odświeżenia

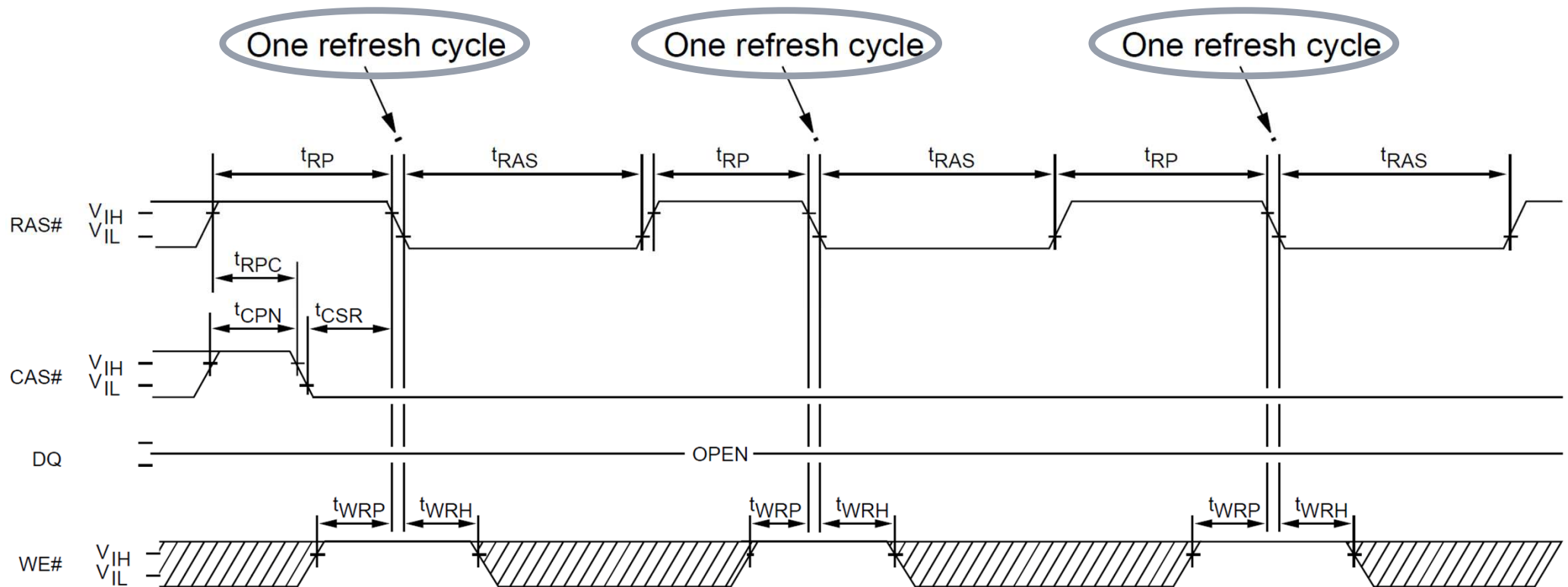


Tryby pracy i odświeżanie pamięci dynamicznych

■ Odświeżanie pamięci dynamiczne

■ Tryb statyczny (ang. CAS before RAS)

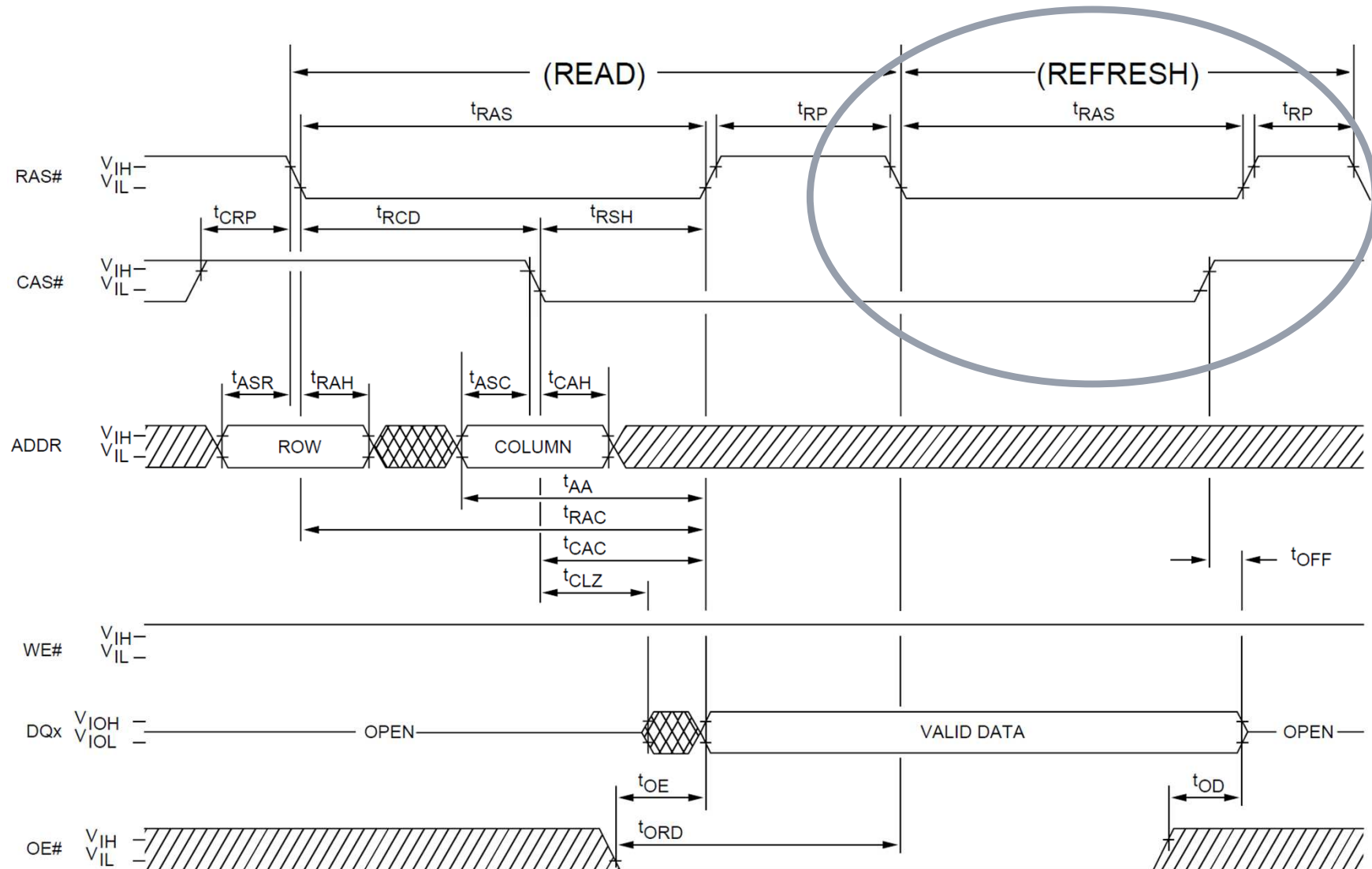
- sekwencja /CAS 1->0 a po niej /RAS 1->0 uruchamia automatyczne odświeżenia wiersza, zgodnie z wewnętrznym licznikiem modułu pamięci



Tryby pracy i odświeżanie pamięci dynamicznych

■ Odświeżanie pamięci dynamiczne

■ Tryb ukryty (ang. Hidden mode)



Tryby pracy i odświeżanie pamięci dynamicznych

- Odkrywanie parametrów modułów pamięciowych
 - Specjalna pamięć zainstalowana w modułach ujawnia ich konfigurację
 - Pamięć EEPROM podłączona za pomocą magistrali I2C

CPU-Z CPU | Caches | Mainboard | Memory | **SPD** | Graphics | Bench | About

Memory Slot Selection

Slot #1 ▼ DDR4 Module Size 8 GBytes

Max Bandwidth DDR4-2666 (1333 MHz) SPD Ext.

Module Manuf. Wilk Elektronik Week/Year 17 / 20

DRAM Manuf. Samsung Ranks Single

Part Number GR2666S464L19S/8G Correction

Serial Number Registered

Timings Table

	JEDEC #11	JEDEC #12	JEDEC #13	JEDEC #14
Frequency	1333 MHz	1333 MHz	1333 MHz	1333 MHz
CAS# Latency	20.0	21.0	22.0	23.0
RAS# to CAS#	19	19	19	19
RAS# Precharge	19	19	19	19
tRAS	43	43	43	43
tRC	61	61	61	61
Command Rate				
Voltage	1.20 V	1.20 V	1.20 V	1.20 V

CPU-Z Ver. 1.94.8.x64 Tools ▼ Validate Close

CPU-Z CPU | Caches | Mainboard | **Memory** | SPD | Graphics | Bench | About

General

Type DDR4 Channel # Dual

Size 16 GBytes DC Mode

Uncore Frequency 498.9 MHz

Timings

DRAM Frequency	1197.4 MHz
FSB:DRAM	1:18
CAS# Latency (CL)	17.0 clocks
RAS# to CAS# Delay (tRCD)	17 clocks
RAS# Precharge (tRP)	17 clocks
Cycle Time (tRAS)	39 clocks
Row Refresh Cycle Time (tRFC)	420 clocks
Command Rate (CR)	2T
DRAM Idle Timer	
Total CAS# (tRDRAM)	
Row To Column (tRCD)	

CPU-Z Ver. 1.94.8.x64 Tools ▼ Validate Close

Tryby pracy i odświeżanie pamięci dynamicznych

■ Budowa przykładowego modułu pamięci (SODIMM)

■ Przykład M471A5244CB0 (4GB, org. 512Mx64)

■ Układy pamięciowe w org. 512Mx16 (78 FBGA)

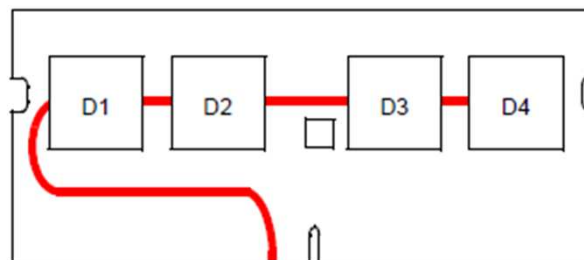
■ DDR4-1600 (**11**-11-11)

tCK(min) 1.25ns, CAS Latency **11**ns, tRCD(min) 13.75ns, tRP(min) 13.75ns

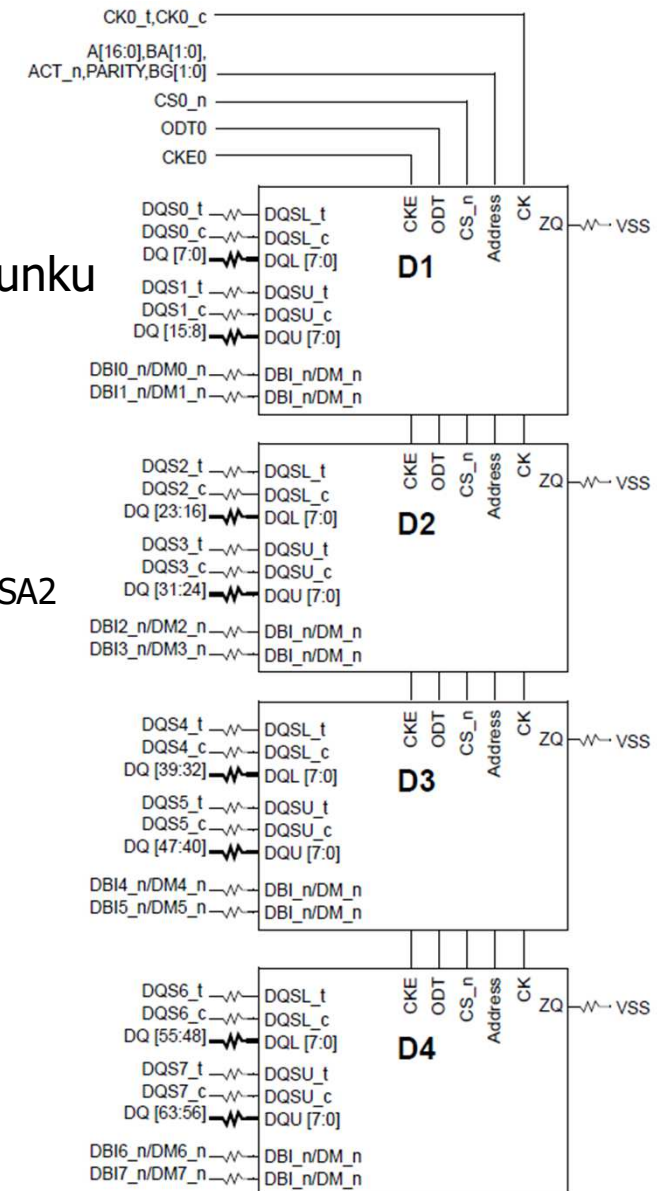
tRAS(min) 35ns, tRC(min) 48.75ns

■ Wyprowadzenia (ważniejsze, część nie pokazana na rysunku obok)

- linie adresowe (multipleksacja): A0–A16, wybór banku: BA0, BA1
- wybór grupy banków: BG0, BG1, Zegarowe: CKE0, CKE1
- sterowania: /RAS_n, /CAS_n, WE_n, CS0_n, CS1_n, PAR
- magistrala danych: DQ0–DQ63, ECC: CB0–CB7
- pamięć konfiguracyjna (tzw. Serial Presence Detect): SCL, SDA, SA0~SA2 (wybór adresu)
- stanu i sterowania dodatkowego: RESET_n, ALERT_n,
- alertów termicznych: EVENT_n
- ...



Address, Command and Control lines



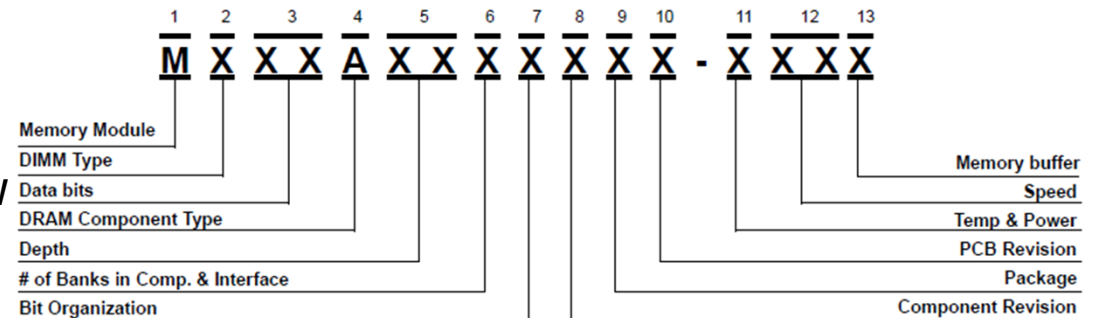
Tryby pracy i odświeżanie pamięci dynamicznych

■ Budowa modułu pamięci (SODIMM), cd.

■ Kodowanie oznaczeń modułów

dla np.: M471A5244CB0

- (M4) typ modułu: SODIMM
- (71) bitów danych: 64bity 260pinów
- (A) komponenty: DDR4 SDRAM
- (52) przestrzeń adr.: 512M
- (4) liczba banków: 16 banków
- (4) organizacja: po 16 bitów
- (C) rewizja: 4th gen.
- (B) obudowa: FBGA
- ...



1. Memory Module : M

2. DIMM Type

- 3 : DIMM
- 4 : SODIMM

3. Data Bits

- 71: x64 260pin Unbuffered SODIMM
- 74: x72 260pin ECC Unbuffered SODIMM
- 78: x64 288pin Unbuffered DIMM
- 86: x72 288pin Load Reduced DIMM
- 91: x72 288pin ECC Unbuffered DIMM
- 92: x72 288pin VLP Registered DIMM
- 93: x72 288pin Registered DIMM

4. DRAM Component Type

- A : DDR4 SDRAM (1.2V VDD)

5. Depth

- | | |
|-----------|-------------------|
| 56 : 256M | AG : 16G |
| 51 : 512M | 1K : 1G (for 8Gb) |
| 1G : 1G | 2K : 2G (for 8Gb) |
| 2G : 2G | 4K : 4G (for 8Gb) |
| 4G : 4G | 8K : 8G (for 8Gb) |
| 8G : 8G | AK : 16G |

6. # of Banks in comp. & Interface

- 4 : 16Banks & POD-1.2V

7. Bit Organization

- 0 : x 4
- 3 : x 8
- 4 : x 16

8. Component Revision

- | | |
|--------------|--------------|
| M : 1st Gen. | A : 2nd Gen. |
| B : 3rd Gen. | C : 4th Gen. |
| D : 5th Gen. | E : 6th Gen. |
| F : 7th Gen. | G : 8th Gen. |

9. Package

- B : FBGA (Halogen-free & Lead-free, Flip Chip)
- M : FBGA (Halogen-free & Lead-free, DDP)
- 2 : FBGA (Halogen-free & Lead-free, 2H TSV)
- 4 : FBGA (Halogen-free & Lead-free, 4H TSV)

10. PCB Revision

- | | |
|--------------|--------------|
| 0 : None | 1 : 1st Rev. |
| 2 : 2nd Rev. | 3 : 3rd Rev. |
| 4 : 4th Rev. | |

11. Temp & Power

- C : Commercial Temp.(0°C ~ 85°C) & Normal Power

12. Speed

- PB : DDR4-2133 (1066MHz @ CL=15, tRCD=15, tRP=15)
- RC : DDR4-2400 (1200MHz @ CL=17, tRCD=17, tRP=17)
- TD : DDR4-2666 (1333MHz @ CL=19, tRCD=19, tRP=19)
- RB : DDR4-2133 (1066MHz @ CL=17, tRCD=15, tRP=15)
- TC : DDR4-2400 (1200MHz @ CL=19, tRCD=17, tRP=17)
- WD : DDR4-2666 (1333MHz @ CL=22, tRCD=19, tRP=19)

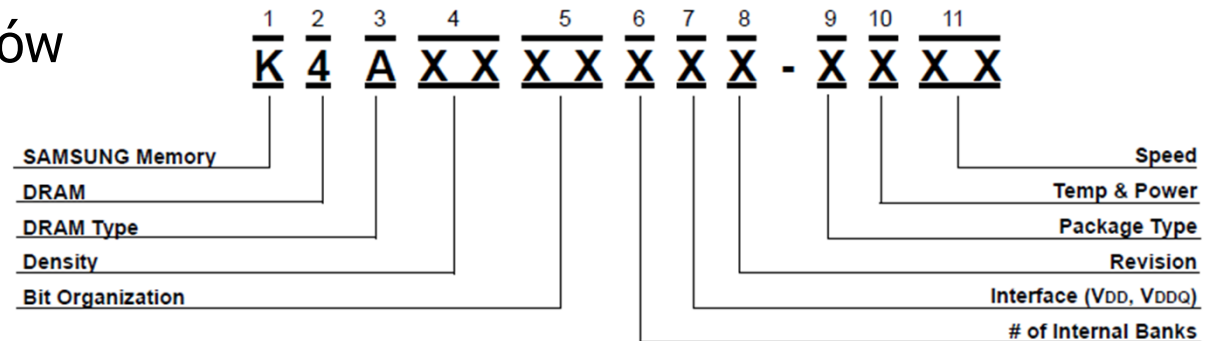
Tryby pracy i odświeżanie pamięci dynamicznych

■ Budowa modułu pamięci (SODIMM), cd.

■ Kodowanie oznaczeń układów

dla np.: K4A8G165WB-xxxx

- (K4A) Samsung DDR4
- (8G) gęstość: 8Gb
- (16) organizacja: x16
- (5) liczba banków: 16banków
- (W) VDD: 1,2V, VDDQ: 1,2V
- (B) rewizja: 3rd Gen
- ...



1. SAMSUNG Memory : K

2. DRAM : 4

3. DRAM Type

A : DDR4 SDRAM (1.2V VDD)

4. Density

4G : 4Gb
8G : 8Gb
AG : 16Gb
BG : 32Gb

5. Bit Organization

04 : x 4
08 : x 8
16 : x16

6. # of Internal Banks

5 : 16 Banks

7. Interface (VDD, VDDQ)

W : POD (1.2V, 1.2V)

8. Revision

M : 1st Gen.
A : 2nd Gen.
B : 3rd Gen.
C : 4th Gen.
D : 5th Gen.
E : 6th Gen.
F : 7th Gen.
G : 8th Gen.

9. Package Type

B : FBGA (Halogen-free & Lead-free, Flip Chip)
M : FBGA (Halogen-free & Lead-free, DDP)
2 : FBGA (Halogen-free & Lead-free, 2H TSV)
4 : FBGA (Halogen-free & Lead-free, 4H TSV)

10. Temp & Power

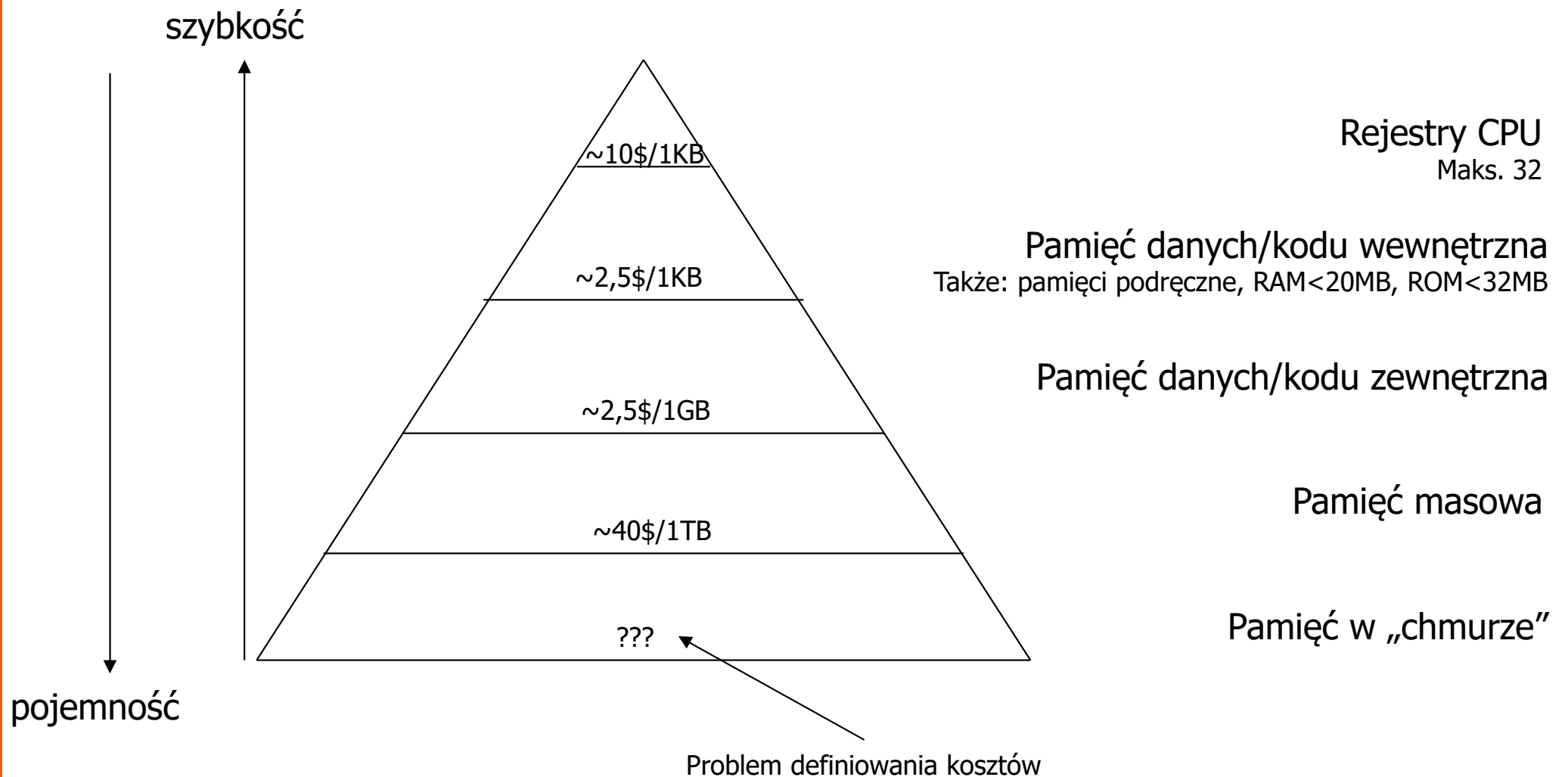
C : Commercial Temp.(0°C ~ 85°C) & Normal Power
I : Industrial Temp.(-40°C ~ 95°C) & Normal Power

11. Speed

PB : DDR4-2133 (1066MHz @ CL=15, tRCD=15, tRP=15)
RC : DDR4-2400 (1200MHz @ CL=17, tRCD=17, tRP=17)
TD : DDR4-2666 (1333MHz @ CL=19, tRCD=19, tRP=19)
RB : DDR4-2133 (1066MHz @ CL=17, tRCD=15, tRP=15)
TC : DDR4-2400 (1200MHz @ CL=19, tRCD=17, tRP=17)
WD : DDR4-2666 (1333MHz @ CL=22, tRCD=19, tRP=19)

Hierarchia pamięci – podsumowanie technologii i umiejscowienia

Hierarchia pamięci – podsumowanie technologii i umiejscowienia

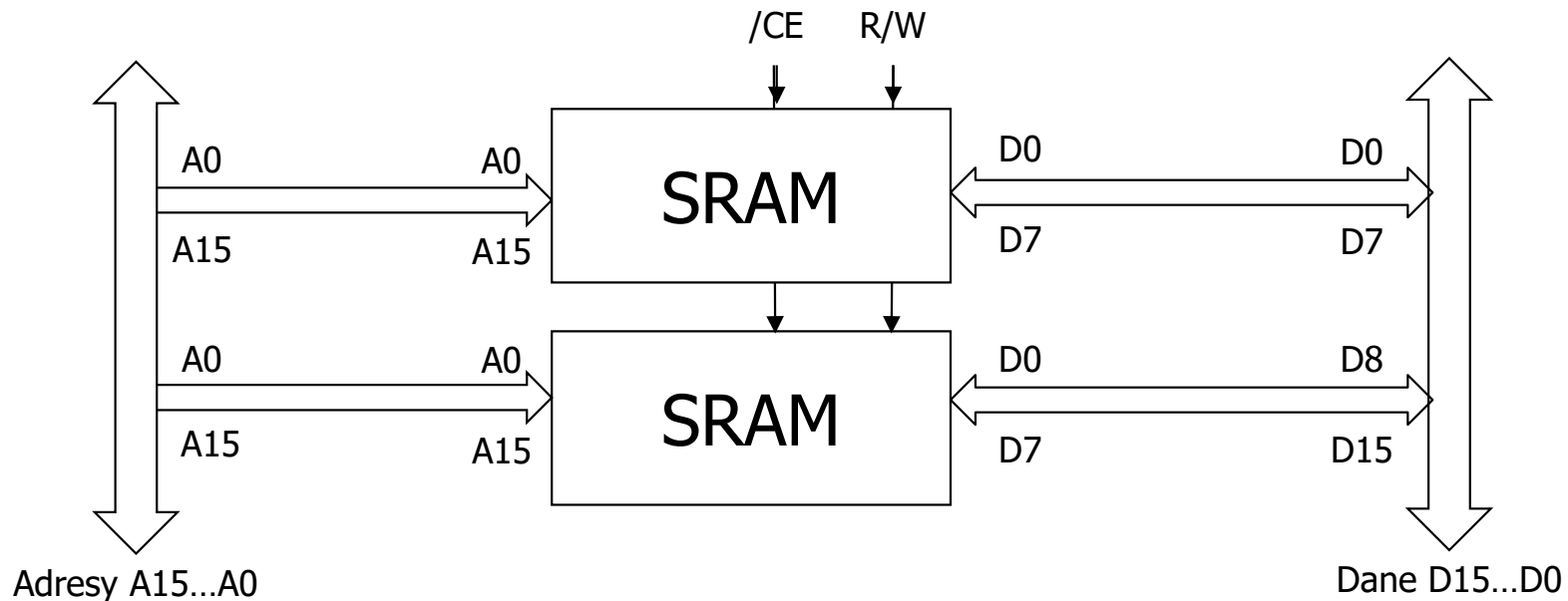


Łączenie pamięci

Łączenie pamięci

■ Zwiększenie szerokości magistrali danych

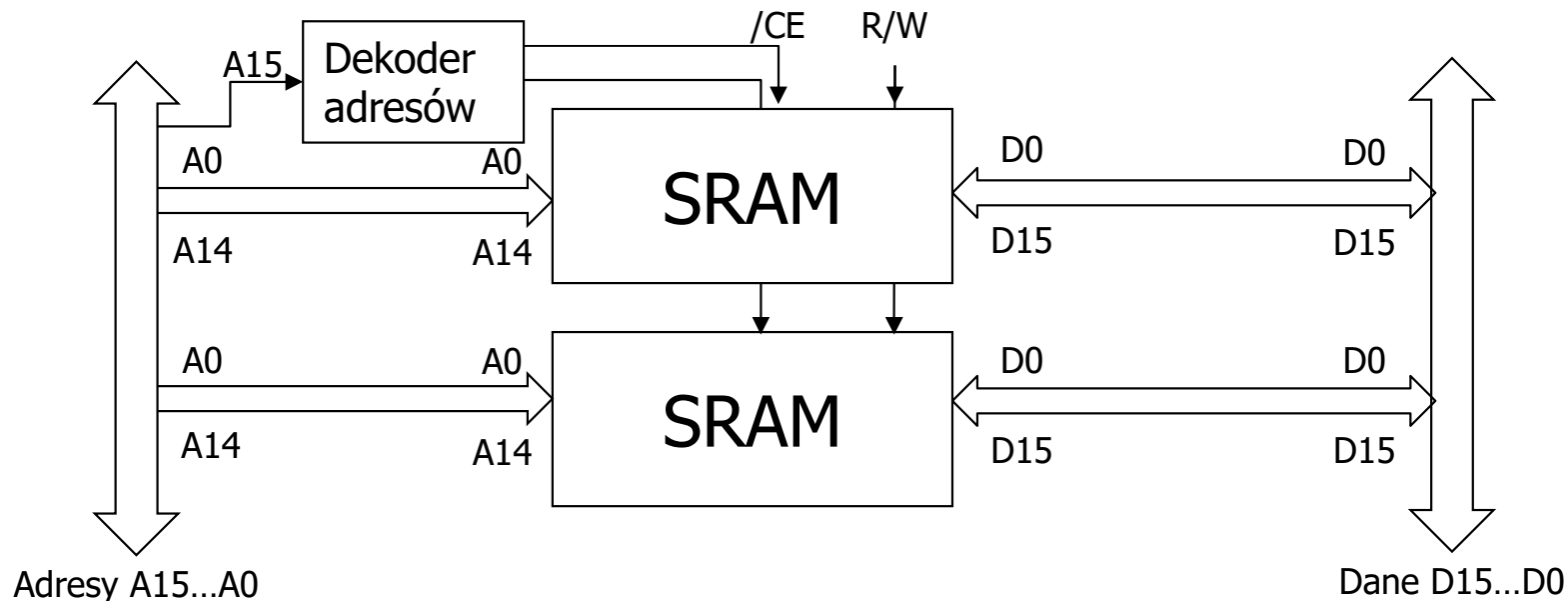
- Warunek: łączone moduły pamięciowe muszą być
 - gdy są tej samej technologii: rodzaj pamięci (SRAM z SRAM, DRAM z DRAM), czas dostępu, ...
 - łączenie pamięci złożonych np.: Flash ROM może być utrudnione – zawierają one specyficzny kontroler który steruje wbudowaną matrycą pamięciową
 - gdy są tej samej wielkości lub wielkość będzie równa wielkości najmniejszego modułu
- Adresy i sterowanie: łączymy równolegle (A0 z A0, ..., A15 z A15, /CE z /CE, ...)
- Dane: rozszerzenie magistrali



Łączenie pamięci

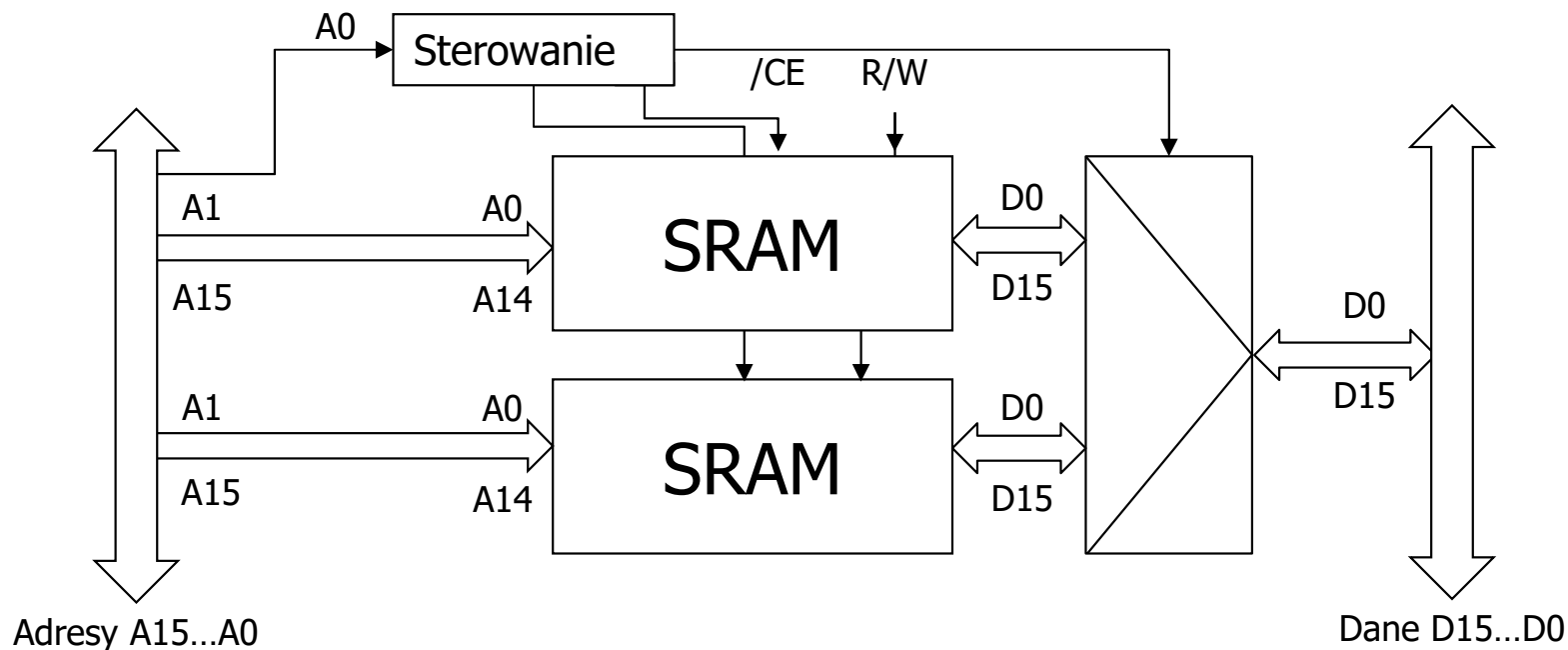
■ Zwiększenie obsługiwanej przestrzeni

- Warunek: jak przy łączeniu dla zwiększenia szerokości magistrali danych
- Dane: łączymy równolegle (D0 z D0, D1 z D1,...)
- Adresy: łączymy równolegle (A0 z A0, A1 z A1,...) brakujące w module pamięci adresy wchodzą na dekodery adresów
 - W tym podejściu dla podanego na magistrali adresowej adresu pracuje tylko jeden moduł pamięci SRAM
- Sterowanie: konieczne wypracowanie sygnałów /CE dla każdego modułu



■ Zwiększenie szybkości

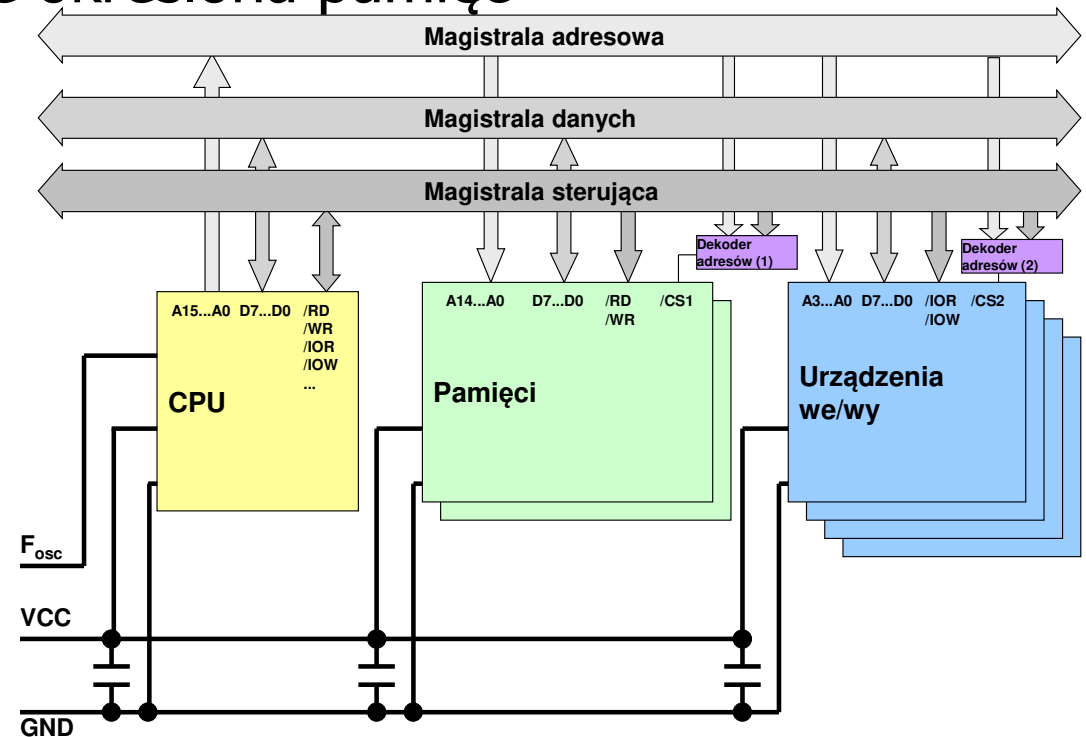
- Możliwe do uzyskania poprzez przekierowanie dostępu do kolejnych lokacji tak aby trafiły do różnych modułów pamięciowych
 - Blok „sterowanie” dekoduje w tym przypadku na bazie A15 który moduł wybrać
 - W tym podejściu oba moduły pamięci SRAM pracują równolegle(!)
 - Tym różni się to podejście od zwykłego zwiększania obsługiwanej przestrzeni (poprzedni slajd)
 - Uwaga! To podejście wprowadza normalny czas dostępu do pierwszej lokacji (wyznacza go czas dostępu górnego modułu) ale otrzymujemy przyspieszenie do odwołania do kolejnej lokacji (dolny moduł zdążył wypracować wynik gdy pierwszy jeszcze pracował)



Dekoder pamięci

Dekoder pamięci

- Dekoder pamięci - jak wspomniano - ustala gdzie w przestrzeni pamięciowej umieszczona będzie określona pamięć



- Założenia dla dalszej analizy

- Architektura pamięci:
- Wielkość magistrali danych CPU:
- Liczba bitów PC:
- Wielkość modułu ROM:
- Wielkość modułu RAM:
- Wektor RESET w CPU pod adresem:
- Brak innych urządzeń w przestrzeni pamięciowej (to może wiele uprościć)

Von Neumana

8bitów

14bitów

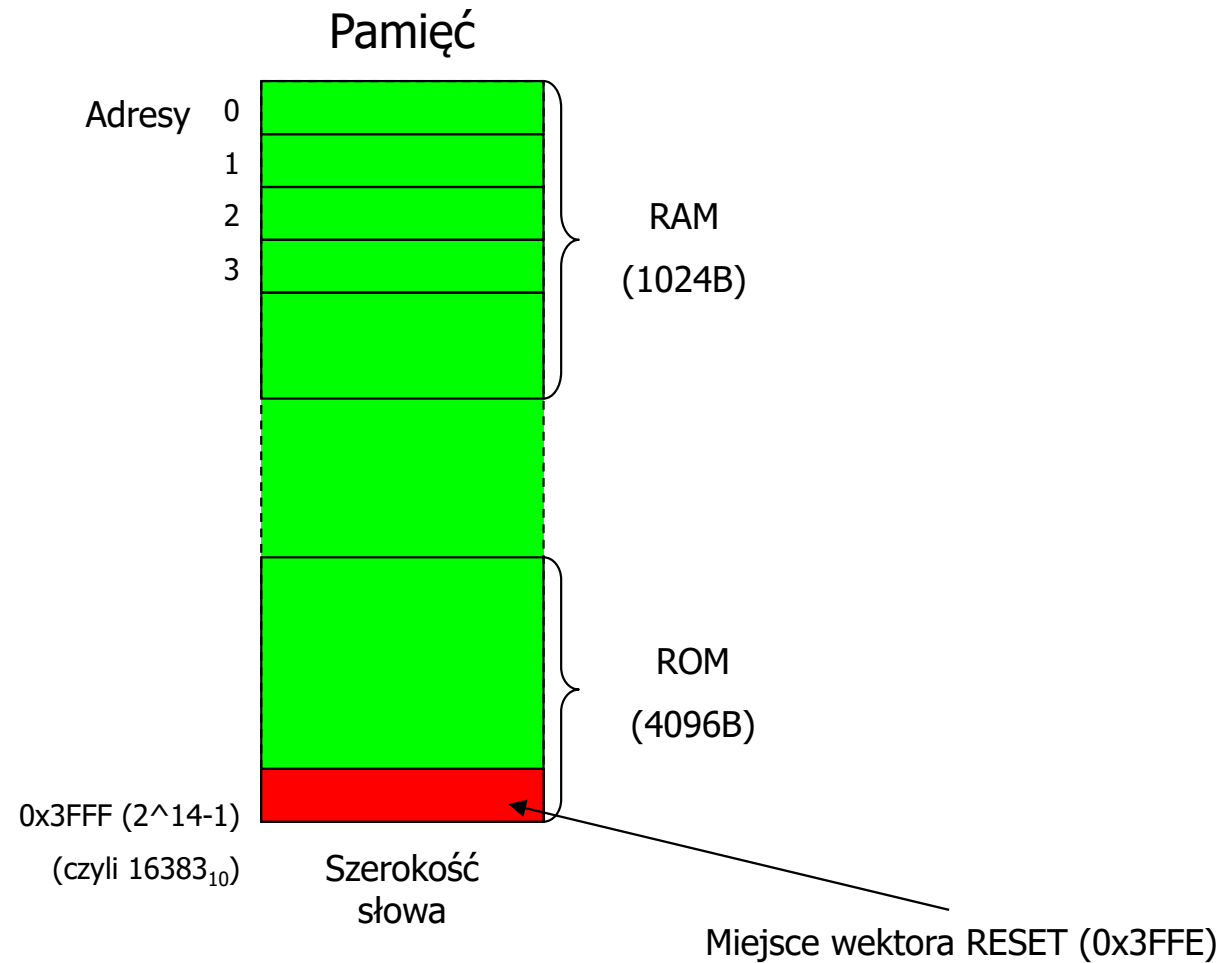
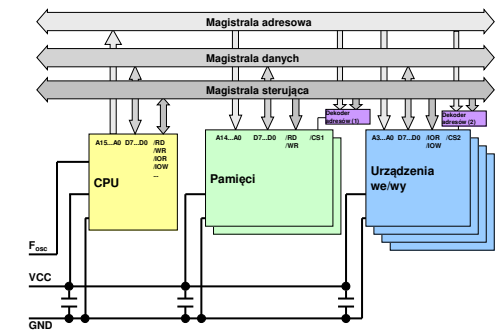
4096B

1024B

0x3FFE (16382)

Dekoder pamięci

- Wynikająca z założeń proponowana mapa pamięci (bez zachowania skali)



Dekoder pamięci

■ Łączenie sygnałów elektrycznych (wynik założeń)

■ CPU

- Adresy: 13...0 (liczba bitów PC)
- Dane: 7...0 (wielkość magistrali danych CPU)
- Sterujące: /RD, /WR

■ ROM

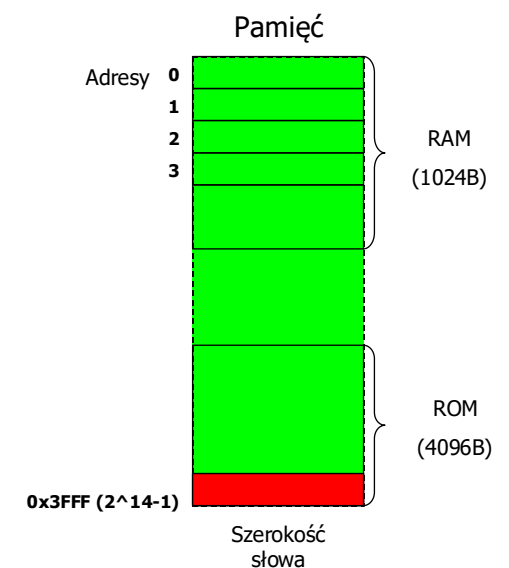
- Adresy: 11...0 (do CPU - adresy: A11-A11, A10-A10, A9-A9, A8-A8, ...)
- Dane: 7...0 (do CPU - dane: D7-D7, D6-D6, ...)
- Sterujące: /CS, /OE

■ RAM

- Adresy: 9...0 (do CPU - adresy: A9-A9, A8-A8, ...)
- Dane: 7...0 (do CPU - dane: D7-D7, D6-D6, ...)
- Sterujące: /CS, /OE, R/W

■ Zatem najprostsz i niedokładny* podział przestrzeni to:

- ROM aktywny gdy A13 i A12 są „1”
- RAM aktywny gdy A13 i A12 są „0”



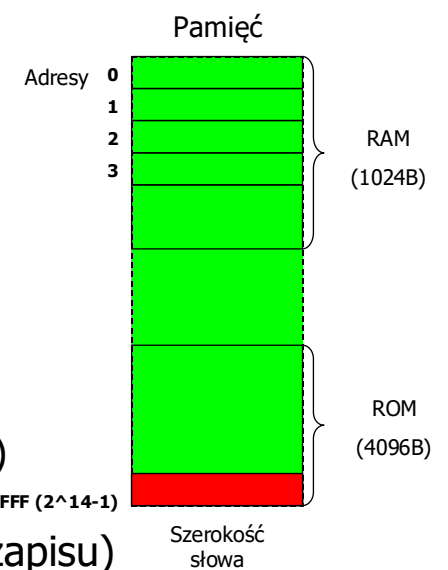
*) Dekodowanie takie stosowane gdy chcemy konstrukcję uprościć (ekonomia) a nie planuje się dodawać do przestrzeni pamięciowej nowych elementów

Dekoder pamięci

■ Tabela prawdy (po wstępnym uproszczeniu)

■ Etap 1: eliminacja stanów nieistotnych (IDLE)

A13	A12	/WR	/RD	/CS _{ROM}	/OE _{ROM}	/CS _{RAM}	/OE _{RAM}	R/W	Opis
1	1	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
1	1	1	0						
1	1	0	1	1	1	1	1	1	IDLE (bo ROM nie ma zapisu)
1	1	0	0						
1	0	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
1	0	1	0						
1	0	0	1	1	1	1	1	1	IDLE (bo ROM nie ma zapisu)
1	0	0	0						
0	1	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
0	1	1	0						
0	1	0	1						
0	1	0	0						
0	0	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
0	0	1	0						
0	0	0	1						
0	0	0	0						



■ Dlaczego mamy stany nieistotne?

■ Brak aktywnych sygnałów /WR i /RD

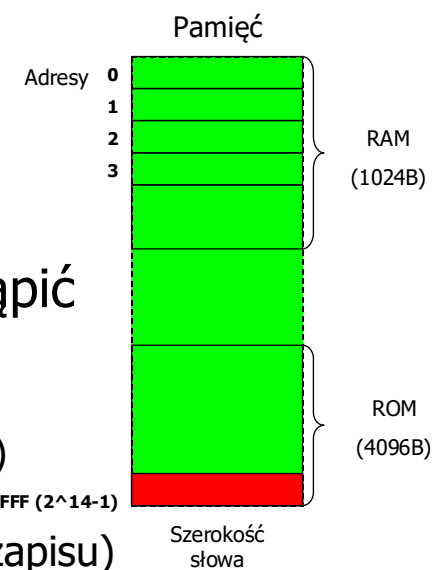
- dla pamięci ROM sygnał /WR także jest nie istotny

Dekoder pamięci

■ Tabela prawdy (po wstępnym uproszczeniu)

- Etap 2: eliminacja stanów sprzecznych – nie mogących wystąpić

A13	A12	/WR	/RD	/CS _{ROM}	/OE _{ROM}	/CS _{RAM}	/OE _{RAM}	R/W	Opis
1	1	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
1	1	1	0						
1	1	0	1	1	1	1	1	1	IDLE (bo ROM nie ma zapisu)
1	1	0	0	1	1	1	1	1	<i>Nie wystąpi!!!</i>
1	0	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
1	0	1	0						
1	0	0	1	1	1	1	1	1	IDLE (bo ROM nie ma zapisu)
1	0	0	0	1	1	1	1	1	<i>Nie wystąpi!!!</i>
0	1	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
0	1	1	0						
0	1	0	1						
0	1	0	0	1	1	1	1	1	<i>Nie wystąpi!!!</i>
0	0	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
0	0	1	0						
0	0	0	1						
0	0	0	0	1	1	1	1	1	<i>Nie wystąpi!!!</i>



- Dlaczego stany nie mogą wystąpić?

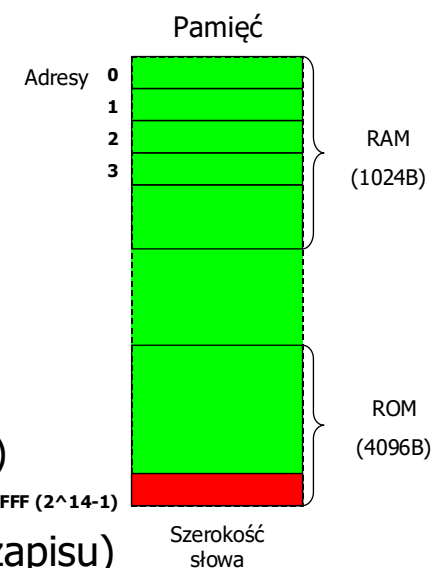
- Specyfika sygnałów sterujących /WR i /RD eliminuje ich równoległą aktywność

Dekoder pamięci

■ Tabela prawdy (po wstępnym uproszczeniu)

■ Etap 3: kiedy pamięć ROM jest aktywna

A13	A12	/WR	/RD	/CS _{ROM}	/OE _{ROM}	/CS _{RAM}	/OE _{RAM}	R/W	Opis
1	1	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
1	1	1	0	0	0	1	1	1	Odczyt z ROM
1	1	0	1	1	1	1	1	1	IDLE (bo ROM nie ma zapisu)
1	1	0	0	1	1	1	1	1	<i>Nie wystąpi!!!</i>
1	0	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
1	0	1	0	0	0	1	1	1	Odczyt z ROM (pozwalamy dla uproszczenia)
1	0	0	1	1	1	1	1	1	IDLE (bo ROM nie ma zapisu)
1	0	0	0	1	1	1	1	1	<i>Nie wystąpi!!!</i>
0	1	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
0	1	1	0						
0	1	0	1						
0	1	0	0	1	1	1	1	1	<i>Nie wystąpi!!!</i>
0	0	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
0	0	1	0						
0	0	0	1						
0	0	0	0	1	1	1	1	1	<i>Nie wystąpi!!!</i>



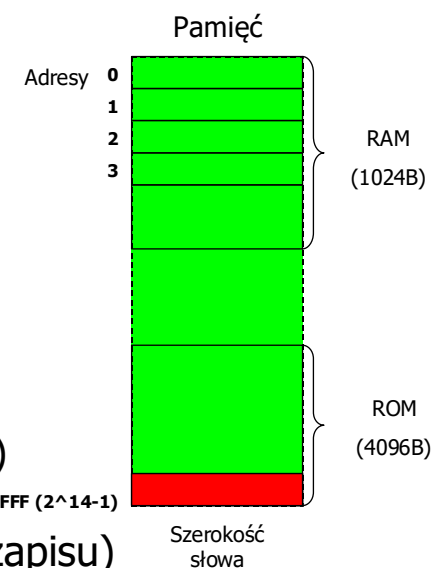
- Dlaczego zakładamy że pamięć ROM może być w dwóch obszarach (A12=0 i A12=1)
 - Aby uprościć dekodowanie – może sygnał A12 będzie w dekodowaniu ROM nie istotny, a dla CPU takie zachowanie nie jest problemem
 - Koszt – brak dekodowania na bazie A12=0 pociąga za sobą w przyszłości brak możliwości dołożenia nowego elementu

Dekoder pamięci

■ Tabela prawdy (po wstępnym uproszczeniu)

■ Etap 4: kiedy pamięć RAM jest aktywna

A13	A12	/WR	/RD	/CS _{ROM}	/OE _{ROM}	/CS _{RAM}	/OE _{RAM}	R/W	Opis
1	1	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
1	1	1	0	0	0	1	1	1	Odczyt z ROM
1	1	0	1	1	1	1	1	1	IDLE (bo ROM nie ma zapisu)
1	1	0	0	1	1	1	1	1	<i>Nie wystąpi!!!</i>
1	0	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
1	0	1	0	0	0	1	1	1	Odczyt z ROM (pozwalamy dla uproszczenia)
1	0	0	1	1	1	1	1	1	IDLE (bo ROM nie ma zapisu)
1	0	0	0	1	1	1	1	1	<i>Nie wystąpi!!!</i>
0	1	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
0	1	1	0	1	1	0	0	1	Odczyt z RAM (pozwalamy dla uproszczenia)
0	1	0	1	1	1	0	1	0	Zapis do RAM (pozwalamy dla uproszczenia)
0	1	0	0	1	1	1	1	1	<i>Nie wystąpi!!!</i>
0	0	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
0	0	1	0	1	1	0	0	1	Odczyt z RAM
0	0	0	1	1	1	0	1	0	Zapis do RAM
0	0	0	0	1	1	1	1	1	<i>Nie wystąpi!!!</i>



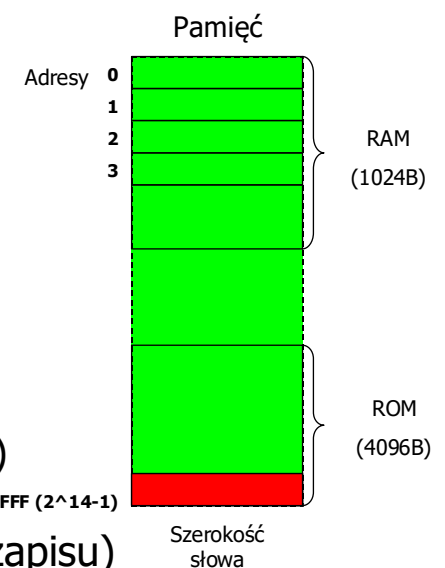
■ Dla RAM możemy stosować te same uproszczenia jak dla ROM

Dekoder pamięci

■ Tabela prawdy (po wstępnym uproszczeniu)

■ Etap 5: uproszczenia zwane minimalizacją

A13	A12	/WR	/RD	/CS _{ROM}	/OE _{ROM}	/CS _{RAM}	/OE _{RAM}	R/W	Opis
1	1	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
1	1	1	0	0	0	1	1	1	Odczyt z ROM
1	1	0	1	1	1	1	1	1	IDLE (bo ROM nie ma zapisu)
1	1	0	0	1	1	1	1	1	<i>Nie wystąpi!!!</i>
1	0	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)
1	0	1	0	0	0	1	1	1	Odczyt z ROM (dla uproszczenia)
1	0	0	1	1	1	1	1	1	IDLE (bo ROM nie ma zapisu)
1	0	0	0	1	1	1	1	1	<i>Nie wystąpi!!!</i>
0	1	1	1	1	1	1	1	1	IDLE (nic się nie dzieje)



Po przekształceniach możemy napisać równania:

$$\text{/CS}_{\text{ROM}} = / (\text{A13} * \text{/WR} * \text{!/RD})$$

$$\text{/OE}_{\text{ROM}} = \text{/CS}_{\text{ROM}}$$

$$\text{/CS}_{\text{RAM}} = / (\text{!A13} * \text{!/WR} * \text{/RD} + \text{!A13} * \text{/WR} * \text{!/RD})$$

$$\text{/OE}_{\text{RAM}} = / (\text{!A13} * \text{/WR} * \text{!/RD})$$

$$\text{/R/W} = / (\text{!A13} * \text{!/WR} * \text{/RD})$$

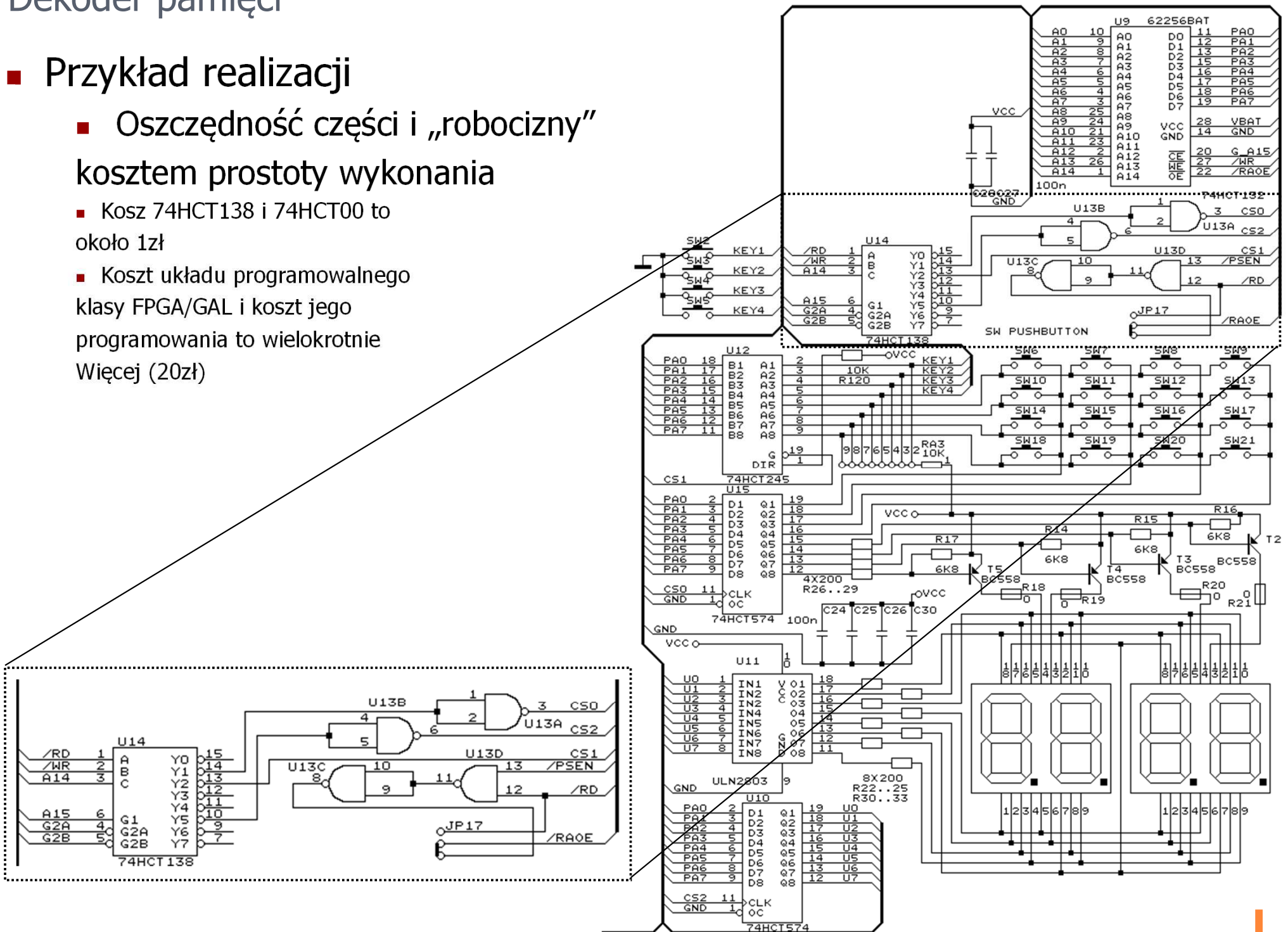
Dekoder pamięci

■ Przykład realizacji

- Oszczędność części i „robocizny”
kosztem prostoty wykonania

- Kosz 74HCT138 i 74HCT00 to
około 1zł

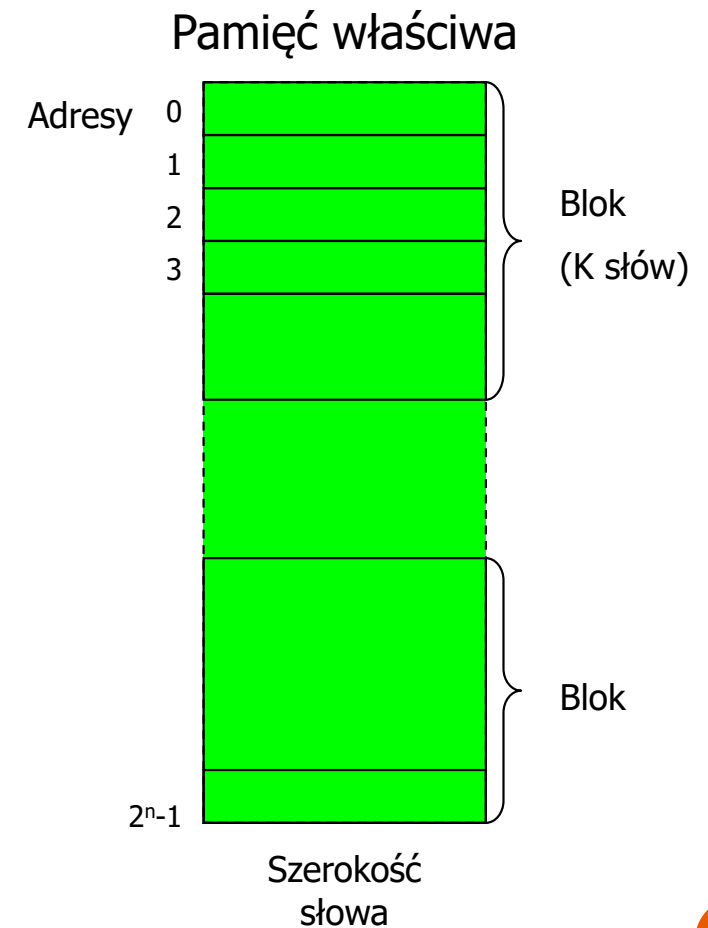
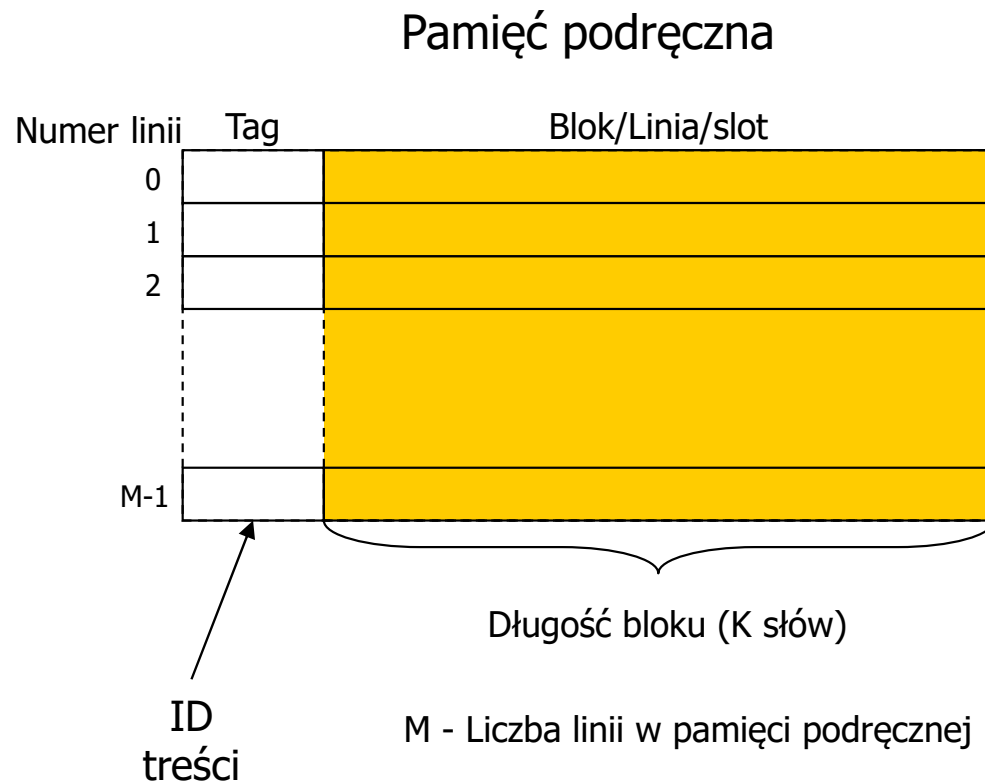
- Koszt układu programowalnego
klasy FPGA/GAL i koszt jego
programowania to wielokrotnie
Wiecej (20zł)



Budowa pamięci podręcznych

Budowa pamięci podręcznych

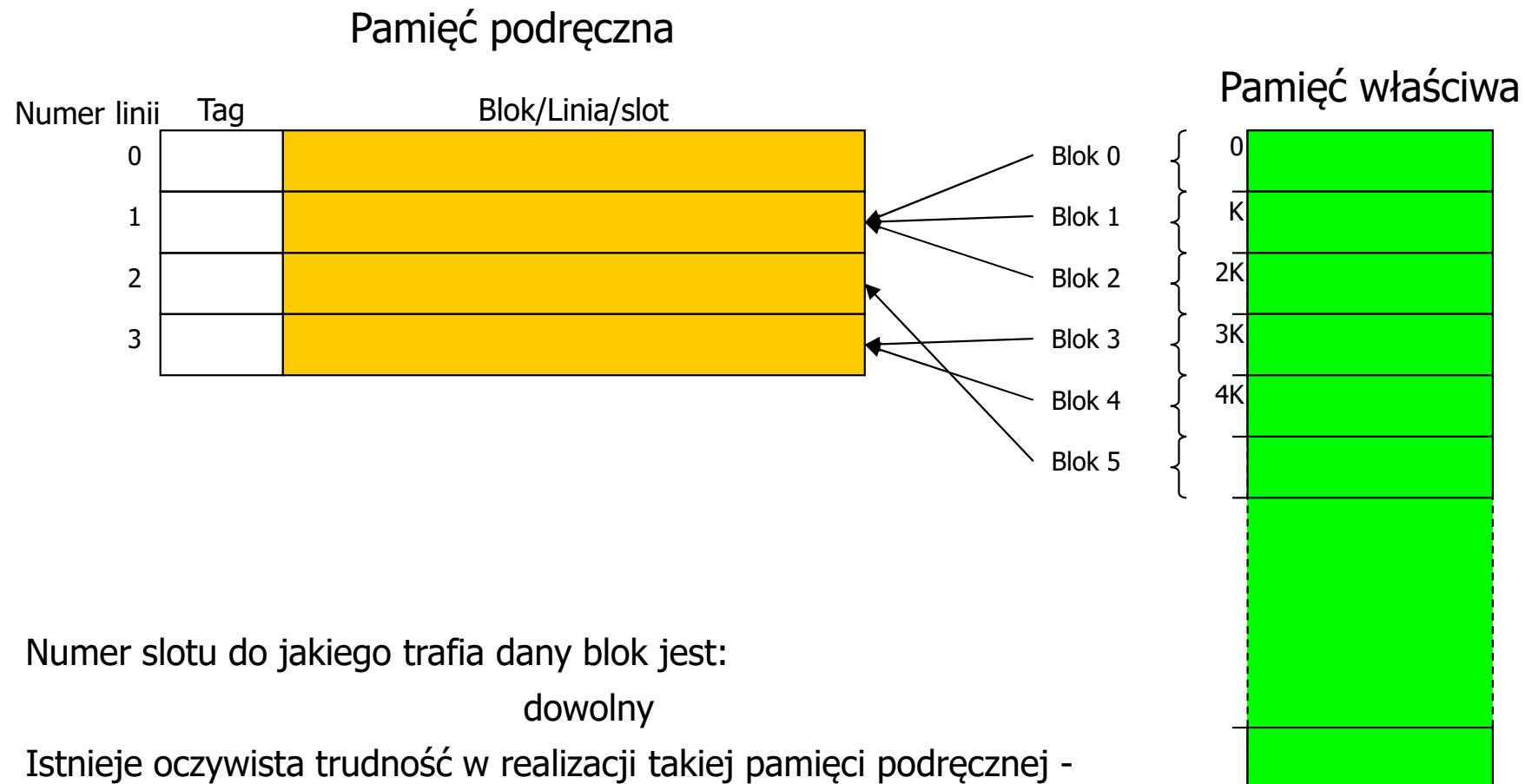
■ Budowa typowych pamięci podręcznych



Budowa pamięci podręcznych

■ Funkcje mapujące

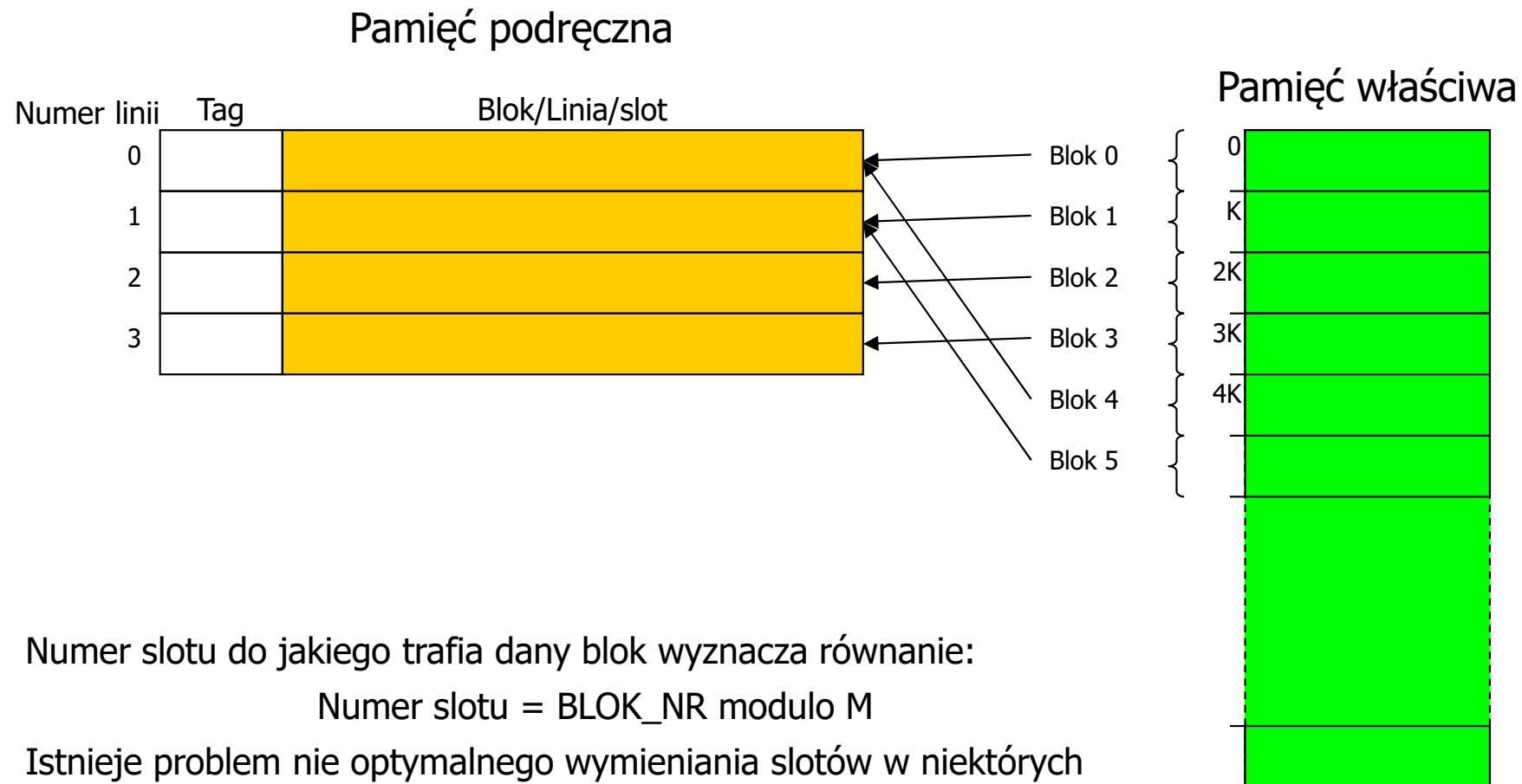
- Asocjacyjne (ang. associative)



Budowa pamięci podręcznych

■ Funkcje mapujące

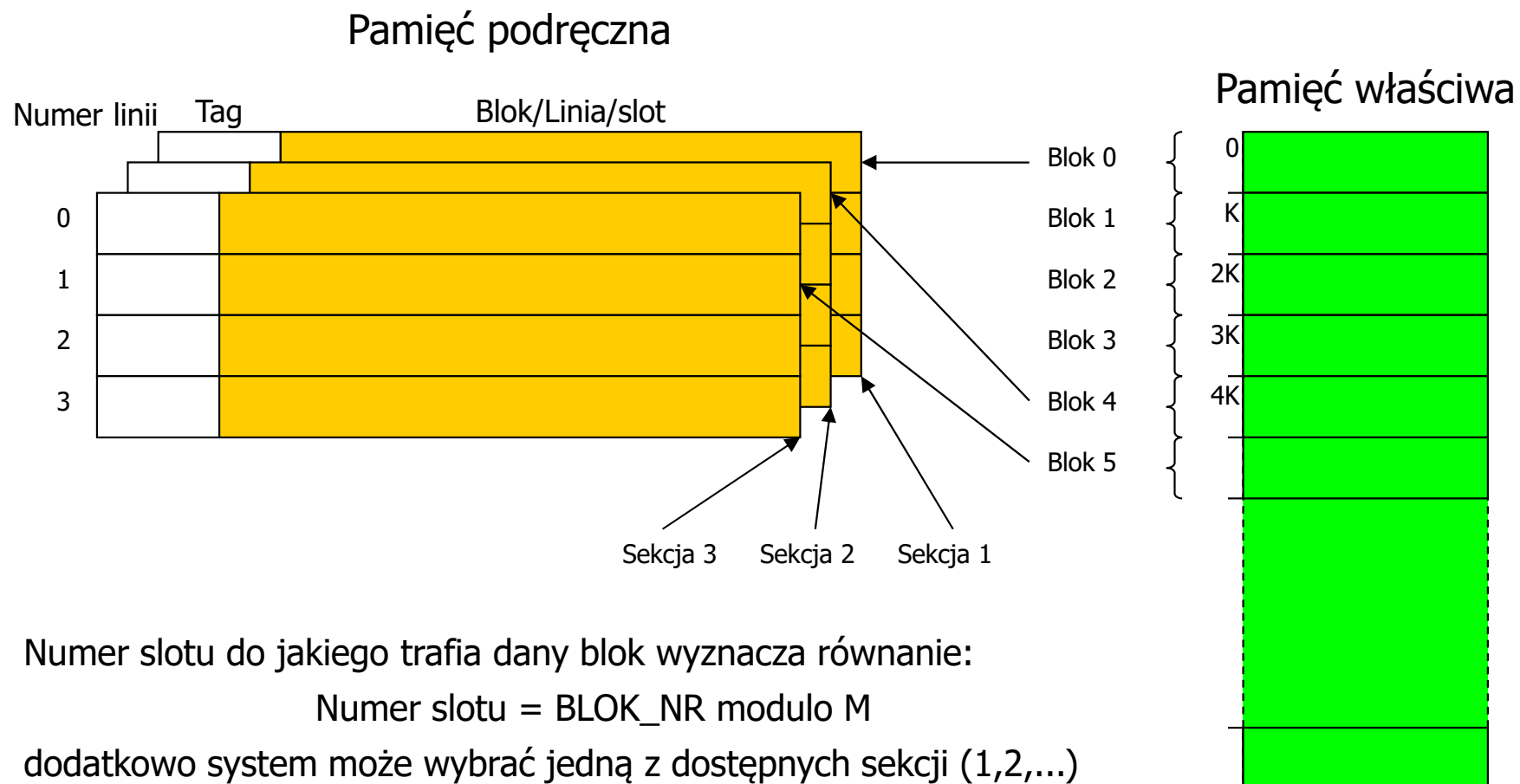
- Bezpośrednie (ang. direct)



Budowa pamięci podręcznych

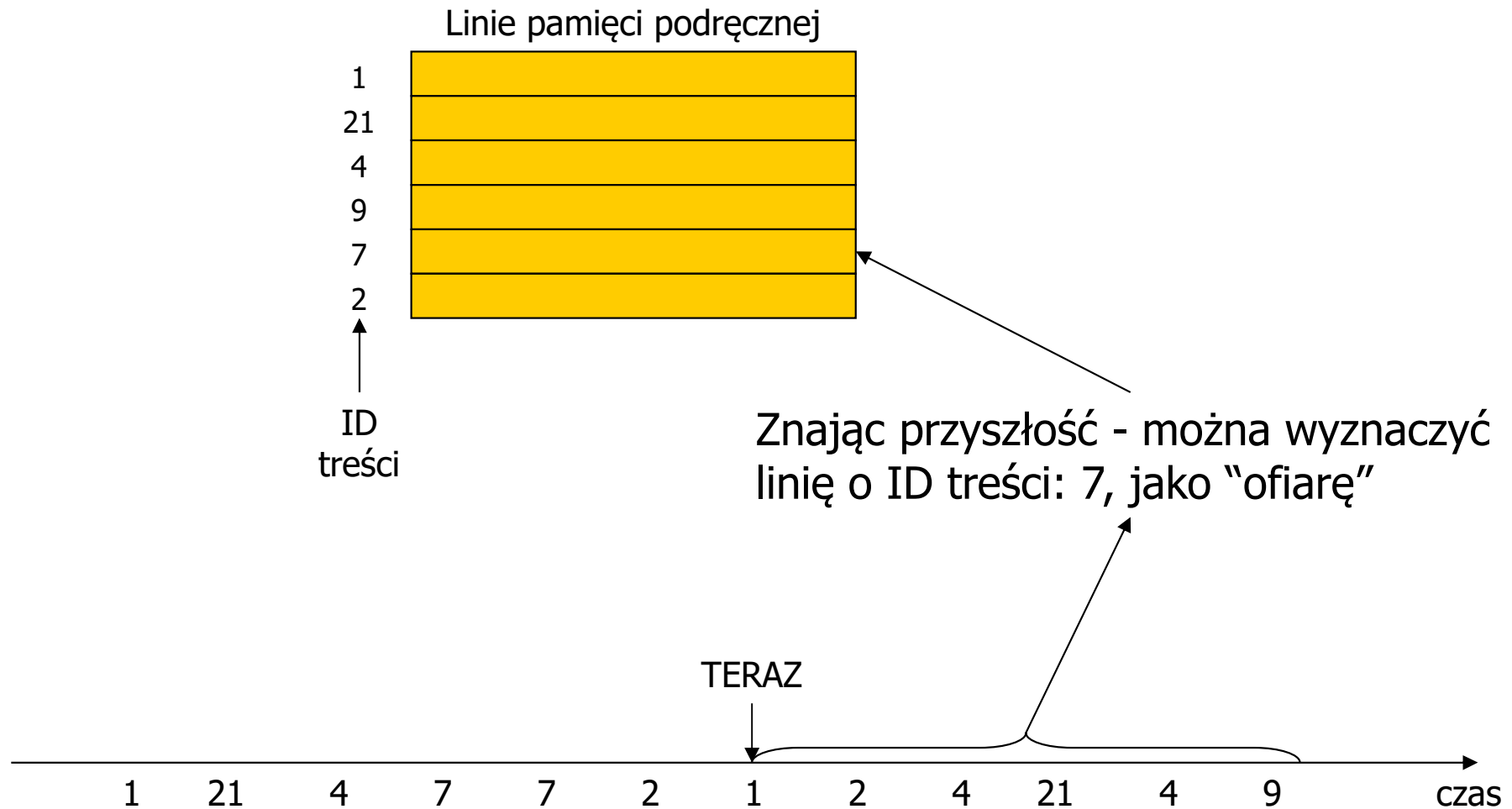
■ Funkcje mapujące

- Asocjacyjne zbiory (ang. set associative, znane jako: n-way)



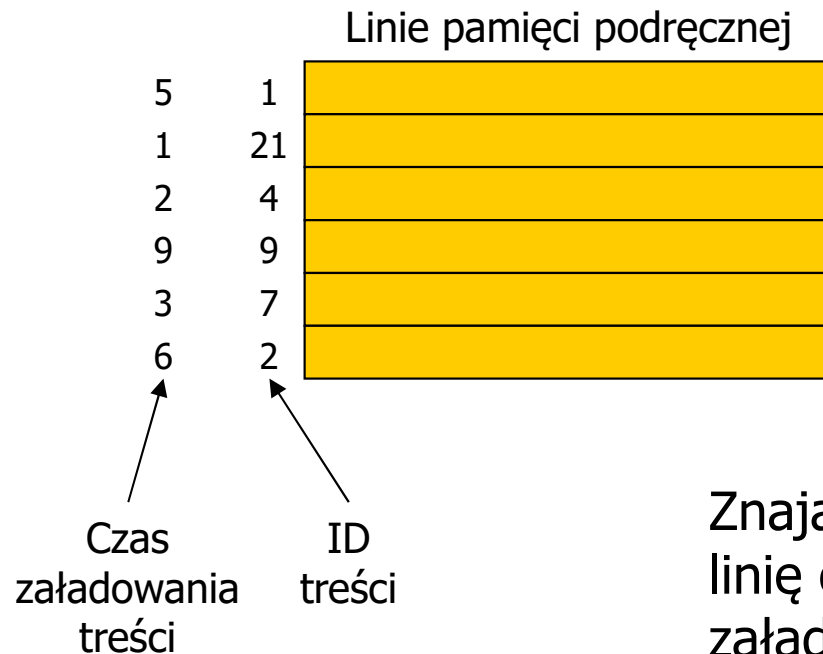
Budowa pamięci podręcznych

- Algorytmy wyboru linii do usunięcia z pamięci podręcznej
 - The Optimal Algorithm (Belady's optimal algorithm)
 - usuwanie linii która przez najdłuższy czas w przyszłości będzie nie potrzebna
 - rozwiązanie wyłącznie teoretyczne



Budowa pamięci podręcznych

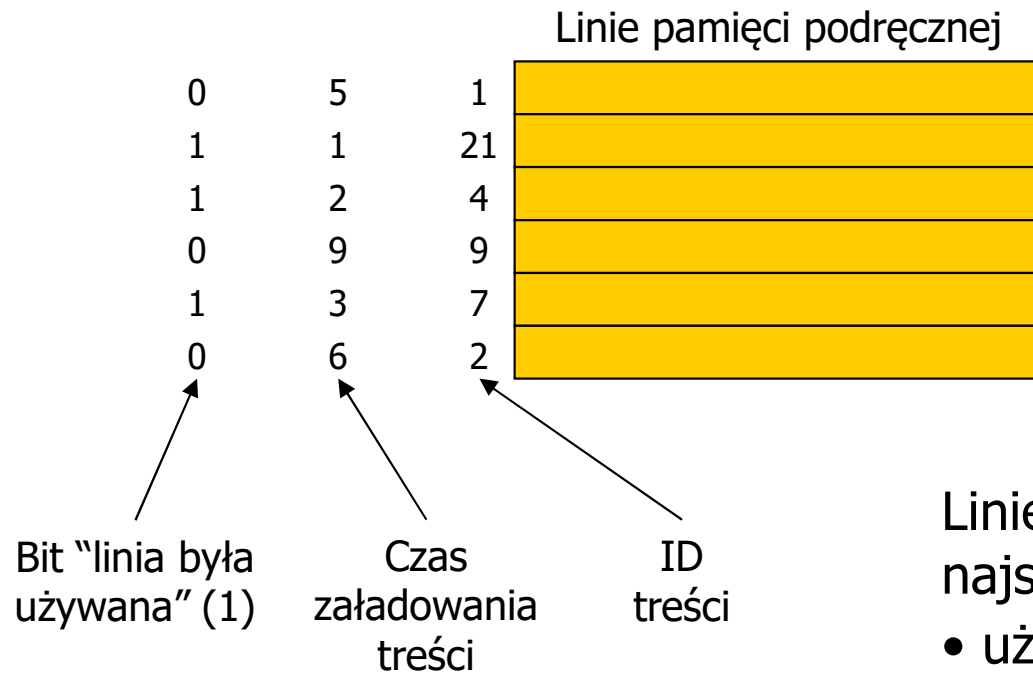
- Algorytmy wyboru linii do usunięcia z pamięci podręcznej
 - First-In-First-Out - FIFO
 - usuwanie linii najdawniej załadowanych



Znając "starość" treści - można wyznaczyć linię o ID treści: 21 jako "ofiare" (jej czas załadowania do pam. podr.: 1)

Budowa pamięci podręcznych

- Algorytmy wyboru linii do usunięcia z pamięci podręcznej
 - Second chance FIFO - "druga szansa dla najstarszych"
 - usuwanie linii najdawniej załadowanych i nieużywanych



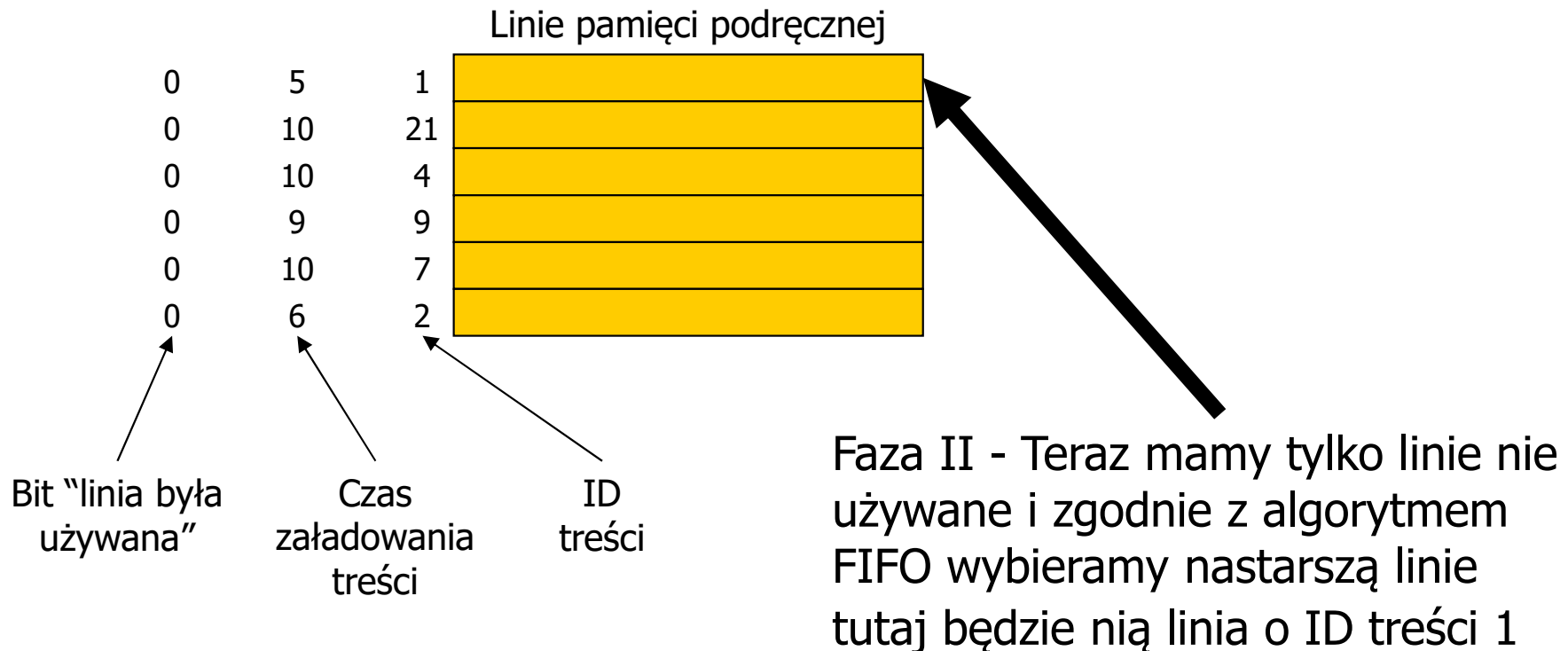
Linie dzielą się (wypisane od najstarszych):

- używane: 1,2,3
- nie używane: 5,6,9

Faza I - Algorytm liniom używanym zmienia "czas załadowania" na czas aktualny (tu: 10) a ich bit użycia na 0

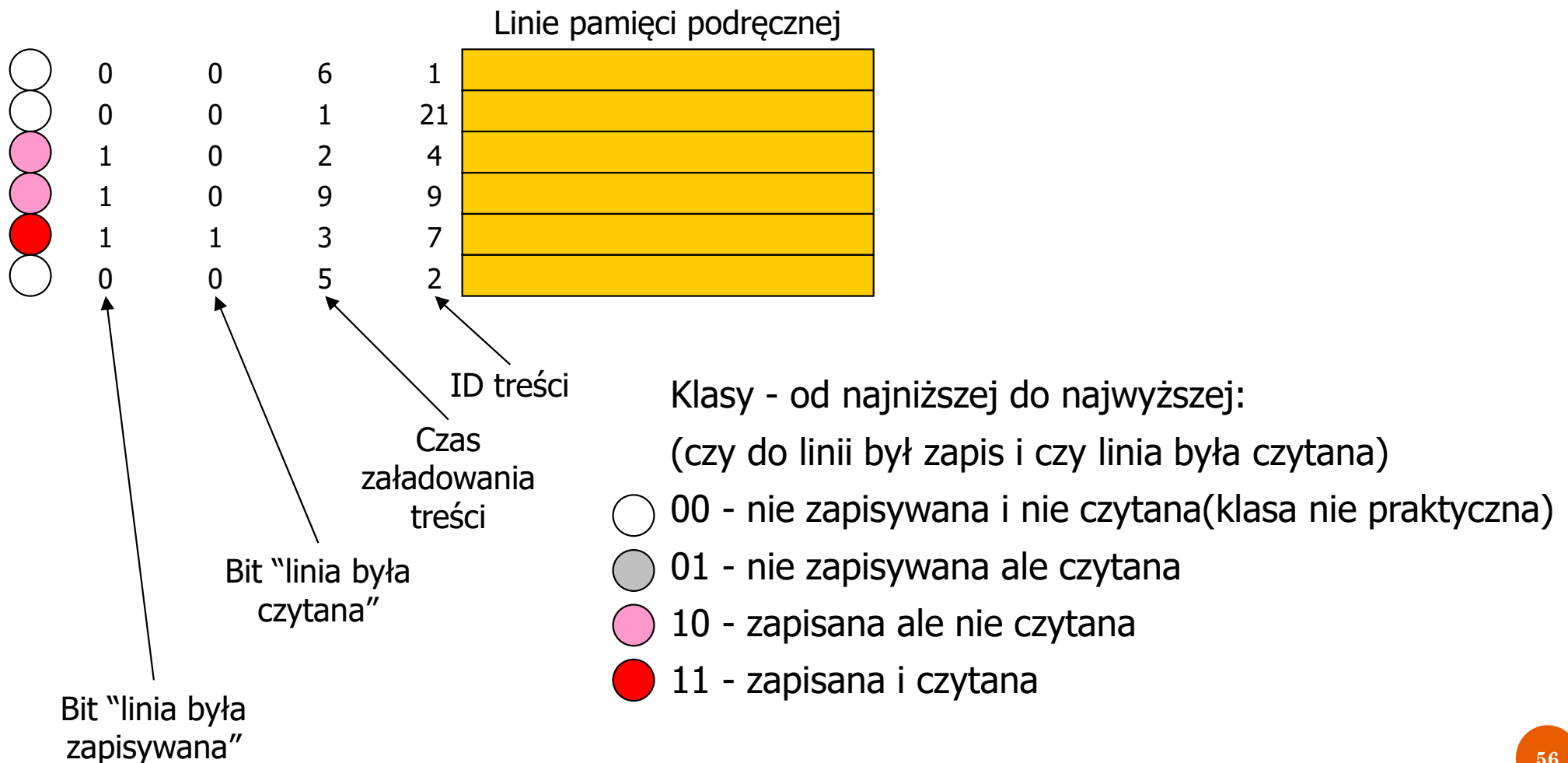
Budowa pamięci podręcznych

- Algorytmy wyboru linii do usunięcia z pamięci podręcznej
 - Second chance FIFO - "druga szansa dla najstarszych", cd.
 - usuwanie linii najdawniej załadowanych i nieużywanych



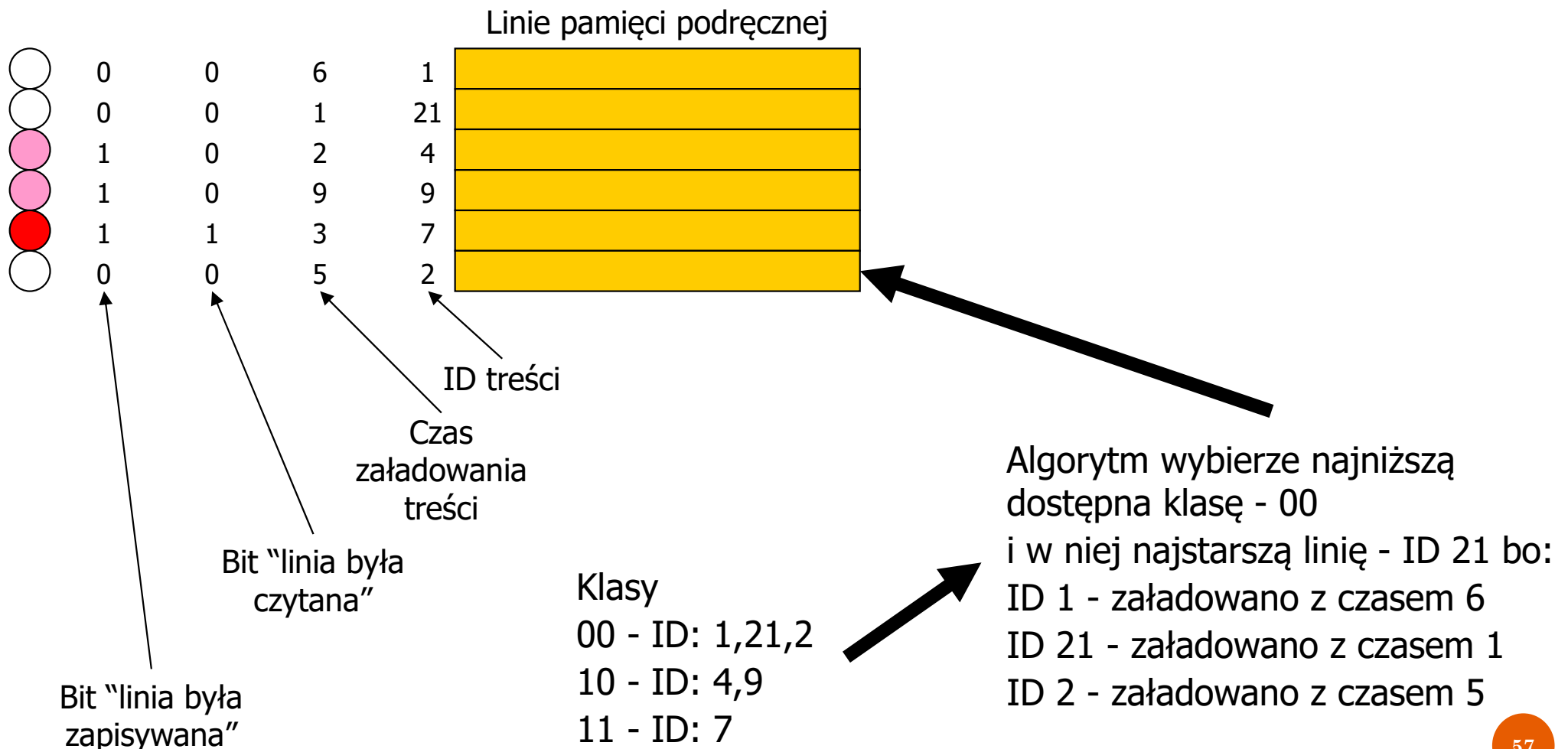
Budowa pamięci podręcznych

- Algorytmy wyboru linii do usunięcia z pamięci podręcznej
 - Enhance second chance FIFO
 - usuwanie linii najdawniej załadowanych z najniższej dostępnej klasy



Budowa pamięci podręcznych

- Algorytmy wyboru linii do usunięcia z pamięci podręcznej
 - Enhance second chance FIFO, cd
 - usuwanie linii najdawniej załadowanych z najniższej dostępnej klasy



Detekcja i korekcja błędów w pamięciach

Detekcja i korekcja błędów w pamięciach

■ Skąd biorą się błędy w pamięciach

- Zakładamy, że treść zapisana została poprawnie – pomijamy znaczenie zapisanej treści i ewentualne błędy tej warstwy
- Źródłami błędów są
 - Problemy podczas transferu do i z pamięci
 - Problemy z utrzymaniem poprawnej treści
 - błędy trwałe
 - uszkodzenie określonych komórek pamięciowych
 - uszkodzenie wewnętrznych połączeń
 - błędy pojawiające się przypadkowo
 - spowodowane niestabilnością napięcia zasilania – tzw. „piki prądowe” (zła filtracja napięć, złe prowadzenie magistral, ...)
 - zakłócenia EMC (ang. Electro Magnetic Compatibility)
 - zakłócenia kosmiczne
 - zużycie się pamięci
 - brak zdolności do poprawnej pracy związany ze starzeniem się elementów (pamięci FLASH ROM, dyski wykorzystujące talerze magnetyczne, pamięci optyczne)

Detekcja i korekcja błędów w pamięciach

■ Jak wykrywać i korygować błędy w pamięciach

■ Testować w procesie produkcji

- usuwanie błędów trwałych

■ Kody zabezpieczające

■ Teoria: odległość Hamminga D_H

- Określa jak daleko są od siebie poszczególne informacje
- Np.: mając liczby 1, 5, 7, 11 – liczba D_H wyznacza najmniejszą odległość (tu różnica) między dwoma dowolnymi liczbami - tu najbliższej siebie są 5 i 7 i odległość wynosi 2

■ Sumy kontrolne

- Umożliwiają detekcję błędów na poziomie nie mniej niż D_H
 - $D_H = 2$ wystarcza do wykrycia przekłamania 1 bitu
 - dla podanego wyżej ciągu liczb pojawienie się wartości 2,3,4,6,8,9,10 są uznane za wartości błędne – ale nie ma metody wyznaczenia jaka ta wartość powinna być
- Realizacja to dodanie pewnej niewielkiej porcji informacji (np. 8bitów) tzw. bitów parzystości (lub nieparzystości) – zapewni D_H stałe i równe 2

Detekcja i korekcja błędów w pamięciach

■ Jak wykrywać i korygować błędy w pamięciach, cd.

■ Kody zabezpieczające, cd.

■ Zabezpieczenia korygujące

- Umożliwiają detekcję błędów na poziomie nie mniej niż $(D_H - 1)/2$

- $D_H > 2$ aby skorygować choć jeden błąd

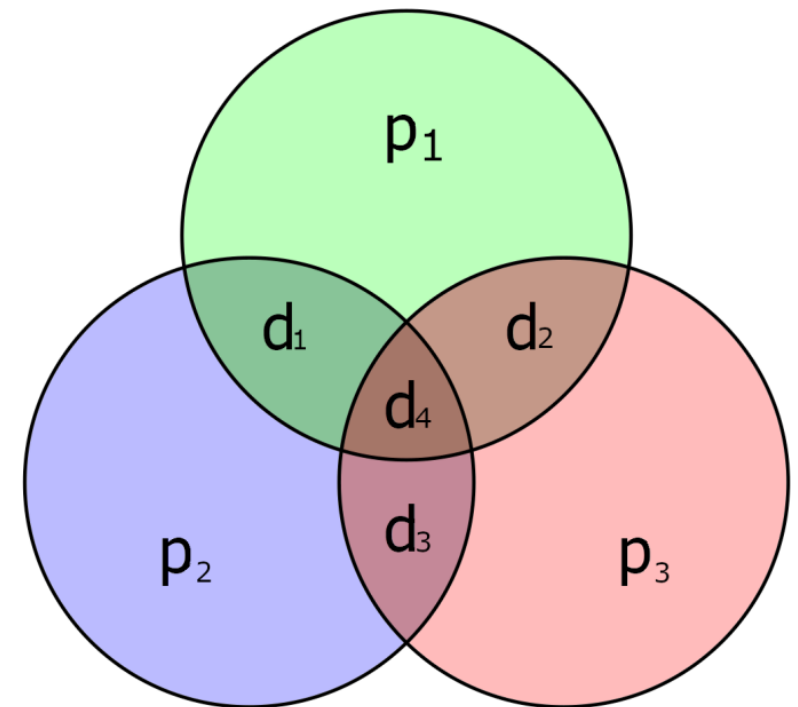
- im D_H większe tym zdolność korekcyjna większa – koszt to mniejsza pojemność tak zabezpieczonej pamięci (lub wymagane dołożenie dodatkowej strefy pamięciowej)

■ Zabezpieczanie z wykorzystaniem kodu Hamminga (7,4)

- 4 bity danych użytkowych - przechowujemy w 7 bitach

- 3 dodatkowe bity błędów wyznaczone są przez obliczenie parzystości z odpowiednich bitów danych

- $P_1 = f(d_1, d_2, d_4)$, $P_2 = f(d_1, d_4, d_3)$, $P_3 = f(d_3, d_4, d_2)$



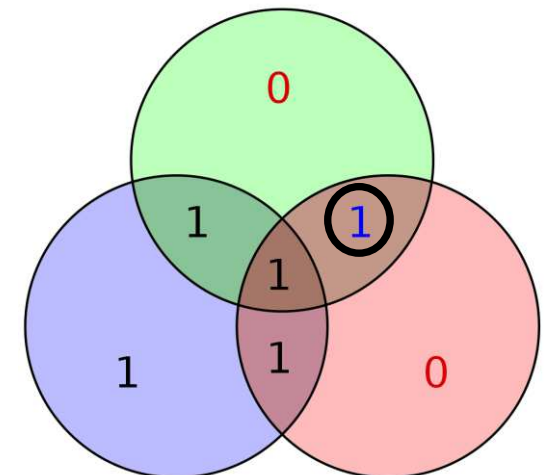
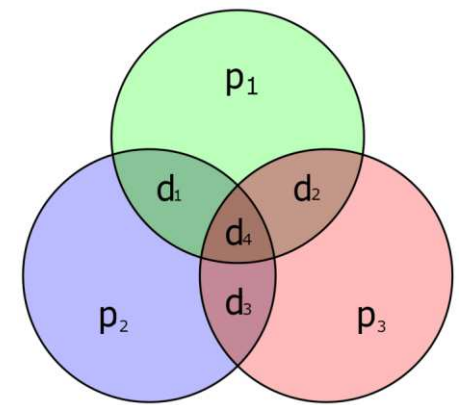
Detekcja i korekcja błędów w pamięciach

■ Jak wykrywać i korygować błędy w pamięciach, cd.

- Zabezpieczanie z wykorzystaniem kodu Hamminga (np.: 7,4), cd.

■ Przykłady korygowania

- odczytano ciąg:
- $(D1, D2, D3, D4, P1, P2, P3) = 1, 1, 1, 1, 0, 1, 0$
- Na bazie D liczymy oczekiwane parzystości
 - $P1 = f(D1, D2, D4) = 1$, $P2 = f(D1, D3, D4) = 1$, $P3 = f(D2, D3, D4) = 1$
- Różnice są dla P1 i P3 – obejmują one część będącą bitem D2 i z tego powodu bit ten powinien być równy 0



■ W pamięciach DDR

- Do 64 bitów danych dodawane jest 8bitów wyliczanych kodem Hamminga

■ W pamięciach FLASH ROM

- Wykonania typu SLC - stosują ECC (Error Correction Code) kod Hamminga
- Wykonania typu MLC - stosują kody blokowe
 - BCH (Bose–Chaudhuri–Hocquenghem)
 - Reed–Solomon

Dziękujemy za uwagę!