

Imię: Kacper

Nazwisko: Kołodyński

Indeks: 249018

Podstawy techniki mikroprocesorowej

Sprawozdanie

Laboratorium 2.4

Prowadzący:

Mgr inż. Wojciech Tarnawski

Zadanie 1

SRAM

Tryb pamięci RAM, statyczna pamięć o dostępie swobodnym. Typ pamięci półprzewodnikowej stosowanej w komputerach, służy jako pamięć buforująca między pamięcią operacyjną i procesorem. Pamięci SRAM wykorzystywane są w szybkich pamięciach podręcznych cache, gdyż nie wymagają one dużych pojemności. Układ posiada dużą szybkość działania i nie wymaga odświeżania

FLASH

Dostęp do danych zapisanych w pamięci flash wykorzystuje stronicowanie pamięci: operacje odczytu, zapisu lub kasowania wykonywane są jednocześnie na ustalonej konstrukcyjnie liczbie komórek, pogrupowanych w strukturę będącą wielokrotnością bajtu. By można było zapisać komórkę pamięci flash, należy ją wcześniej skasować. Nie jest możliwe ponowne zapisanie danych do już zapisanej komórki. Jakkolwiek można odczytać i zapisać dowolną komórkę pamięci, to operacja kasowania umożliwia skasowanie tylko całych bloków komórek. Nie można skasować pojedynczej komórki. Z tego powodu zapis danych nie jest w pełni swobodny. Pamięci te umożliwiają odczyt i zapis dowolnej komórki, ale już nie swobodne kasowanie i nadpisanie zawartości.

Układy FLASH wymagają specjalnych procedur dostępu oraz dodatkowych zmiennych (buforów) do przechowywania większych ilości danych (a co za tym idzie więcej pamięci RAM mikrokontrolera), aby podczas zmiany kilku bajtów w pamięci nie utracić pozostałych danych przechowywanych na tej samej stronie/sektorze pamięci.

EEPROM

Mieści zazwyczaj bardzo niewielką ilość danych, które pozostają zapisane nawet po zaniku zasilania. Należy mieć na uwadze, że liczba zapisów oraz usunięć danych na EEPROM jest ograniczona, natomiast liczba odczytów jest nieskończona. Proces zapisania lub nadpisania może dotyczyć pojedynczego bajta (nie jak w przypadku pamięci flash, gdzie czyszczenie i nadpisywanie należy wykonywać zbiorczo)

Stosowana jest do przechowywania małej ilości danych, które muszą być dostępne po zaniku zasilania.

Rozwinięciem technologii pamięci EEPROM jest pamięć flash, w której ponadto dzięki zastosowaniu buforów zwiększono szybkość zapisu do pamięci.

Atmega32a posiada:

1024 bajty pamięci EEPROM

Zalety

- Możliwość zapisu pojedynczych bajtów (mniejsza konsumpcja pamięci RAM w mikrokontrolerze)

Wady

- Próg pojemności 1Mbit
- Niska prędkość transmisji

32 kilobajty pamięci Flash (32000)

Zalety:

- Szybkość zapisu i odczytu danych
- Zapis danych większymi partiami (dodatkowy RAM na bufor) (wyższy transfer danych niż w EEPROM)
- Rozmiary aż do 4Mbit

Wady:

- Brak możliwości zmieniania i nadpisywania istniejących danych.
- Limit cykli

2048 bajtów wewnętrznej pamięci danych SRAM

Zalety:

- Szybka
- Wydajna

Wady:

- Mniejsza pojemność
- Wysoka cena

Różnica między pamięciami EEPROM i FLASH jest taka, że w układach EEPROM mamy możliwość zapisywania(nadpisywania) pojedynczych bajtów, zaś w układach FLASH czyszczenie oraz zapis trzeba przeprowadzać zbiorowo.

Układy EEPROM z kolei są częściej wykorzystywane tam gdzie wymagany jest sekwencyjny zapis pojedynczymi bajtami. W przypadku układów FLASH przed zapisem danych wymagane jest wykonanie procedury czyszczenia pamięci, a minimalny obszar jaki możemy wyczyścić to pojedynczy sektor. Sekwencja zapisu kilku bajtów do pamięci FLASH powinna być następująca:

1. odczyt danych z sektora do bufora,
2. wyczyszczenie sektora w pamięci
3. modyfikacja bajtów w buforze,
4. zapis danych do sektora.

W przypadku układów **EEPROM** możemy czyścić oraz zapisywać nawet pojedyncze bajty w pamięci. Niestety, tego typu pamięci są wolniejsze zarówno w odczycie jak i zapisie od FLASH.

1.

Wynik kompilacji kodu dla tablicy jednoelementowej.

```
-----  
Device: atmega32  
  
Program:      424 bytes (1.3% Full)  
(.text + .data + .bootloader)  
  
Data:         4 bytes (0.2% Full)  
(.data + .bss + .noinit)
```

Finished building: sizedummy

Program znajduje się w pamięci Flash natomiast dane (tablica jednoelementowa) w pamięci SRAM

2.

Wynik kompilacji kodu dla tablicy stu elementowej.

```
Device: atmega32  
  
Program:      424 bytes (1.3% Full)  
(.text + .data + .bootloader)  
  
Data:        400 bytes (19.5% Full)  
(.data + .bss + .noinit)
```

Tablica przechowywana jest w pamięci SRAM

3.

Wynik kompilacji kodu dla tablicy 1000-elementowej. Tablica jest za duża by przechowywać w wybranym mikrokontrolerze. Pamięć SRAM mikrokontrolera atmega32 to 2048 bajtów, w tym przypadku potrzebujemy ich 4000.

Device: atmega32

Program: 424 bytes (1.3% Full)
(.text + .data + .bootloader)

Data: 4000 bytes (195.3% Full)
(.data + .bss + .noinit)

4.

Tablica 1000 elementowa w pamięci FLASH. Jesteśmy w stanie uruchomić program. Do programu można zapisywać zmienne.

Device: atmega32

Program: 4408 bytes (13.5% Full)
(.text + .data + .bootloader)

Data: 0 bytes (0.0% Full)
(.data + .bss + .noinit)

Zadanie 2

Tab jest zapisana w pamięci FLASH jest to możliwe dzięki „PROGMEM”

Tab2 jest zapisana w pamięci SRAM, jest to domyślna pamięć gdzie zapisujemy zmienne.

```
const uint8_t tab[10] PROGMEM = { 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 };
const uint8_t tab2[10] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };

int main() {

    LCD_Inititalize();
    LCD_Home();
    char text[20];

    uint8_t zmienna1=0;
    //uint8_t zmienna2=0;
    int i = 0;
    while (1)
    {
        zmienna1=pgm_read_byte(&tab[i]);

        LCD_Clear();
        LCD_GoTo(0, 0);
        sprintf(text, "Z1: %d", zmienna1);
        LCD_WriteText(text);

        _delay_ms(250);
        i++;
        if (i == sizeof(tab)/sizeof(tab[0]))
        {
            i = 0;
        }
    }
}
```