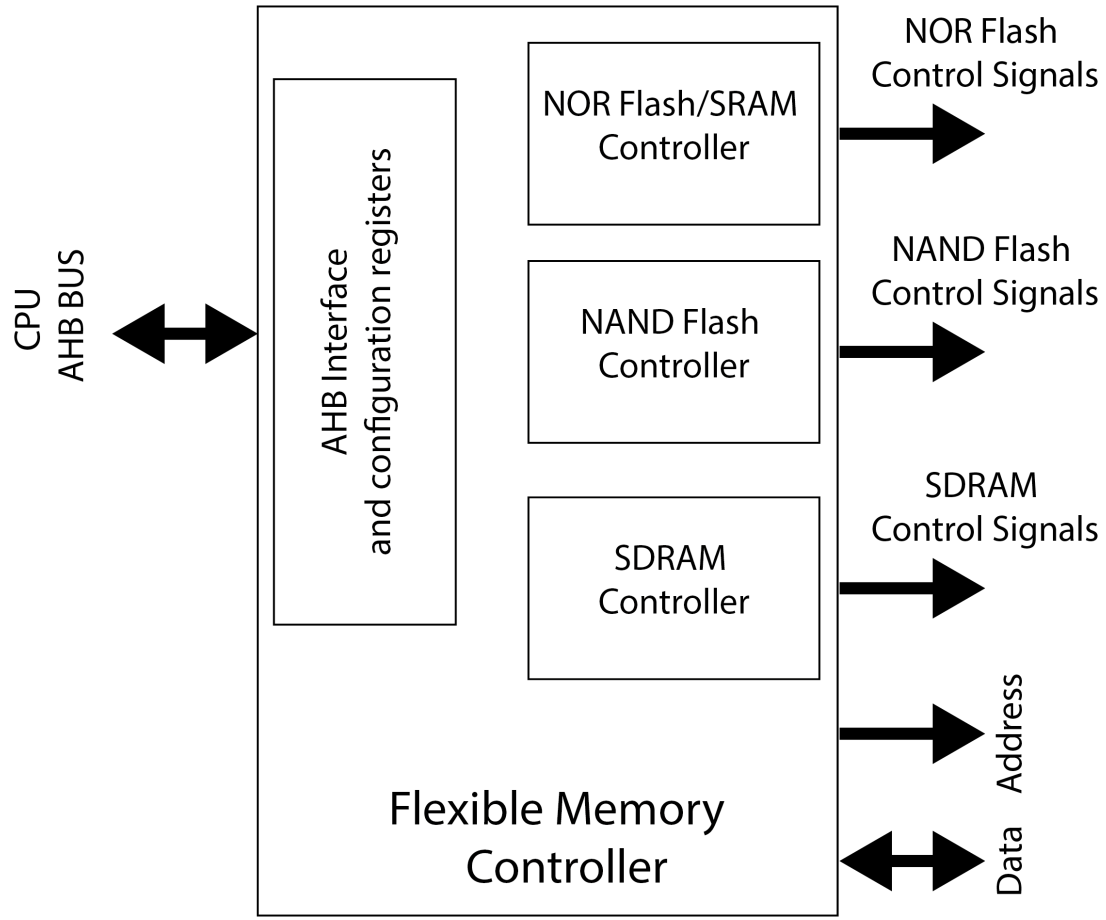
Flexible Memory Controller & FLASH

Rok Češnovar

ORS

Flexible Memory Controller



STM32F407VGT6

- 1MB NAND Flash razdeljen v 12 sektorjev
  - Sector 2 - začetek uporabniškega (podatkovnega) dela

Table 5. Flash module organization (STM32F40x and STM32F41x)

Block	Name	Block base addresses	Size
Main memory	Sector 0	0x0800 0000 - 0x0800 3FFF	16 Kbytes
	Sector 1	0x0800 4000 - 0x0800 7FFF	16 Kbytes
	Sector 2	0x0800 8000 - 0x0800 BFFF	16 Kbytes
	Sector 3	0x0800 C000 - 0x0800 FFFF	16 Kbytes
	Sector 4	0x0801 0000 - 0x0801 FFFF	64 Kbytes
	Sector 5	0x0802 0000 - 0x0803 FFFF	128 Kbytes
	Sector 6	0x0804 0000 - 0x0805 FFFF	128 Kbytes
	. . .	. . .	. . .
	Sector 11	0x080E 0000 - 0x080F FFFF	128 Kbytes
System memory		0x1FFF 0000 - 0x1FFF 77FF	30 Kbytes
OTP area		0x1FFF 7800 - 0x1FFF 7A0F	528 bytes
Option bytes		0x1FFF C000 - 0x1FFF C00F	16 bytes

Pisanje podatkov v Flash

1. odkleni Flash
2. briši sektor (ali celoten Flash)
3. zapiši nove vrednosti
4. zakleni Flash

Pisanje podatkov v Flash

1. odkleni Flash

```
HAL_FLASH_Unlock();
```

2. briši sektor ciljnega naslova (ali celoten Flash)

```
uint32_t SectorError = 0;
FLASH_EraseInitTypeDef EraseInitStruct;

EraseInitStruct.TypeErase = FLASH_TYPEERASE_SECTORS; // FLASH_TYPEERASE_MASSERASE
EraseInitStruct.VoltageRange = FLASH_VOLTAGE_RANGE_3;
EraseInitStruct.Sector = ID_SEKTORJA;
EraseInitStruct.NbSectors = ST_SEKTORJEV;

HAL_FLASHEx_Erase(&EraseInitStruct, &SectorError);
```

Pisanje podatkov v Flash

3. zapiši nove vrednosti (4 bajte)

```
HAL_FLASH_Program(FLASH_TYPEPROGRAM_WORD, naslov, podatek);
```

4. zakleni Flash

```
HAL_FLASH_Lock();
```

Vklop/izklop/brisanje predpomnilnika

```
__HAL_FLASH_DATA_CACHE_DISABLE();
__HAL_FLASH_INSTRUCTION_CACHE_DISABLE();

__HAL_FLASH_DATA_CACHE_RESET();
__HAL_FLASH_INSTRUCTION_CACHE_RESET();

__HAL_FLASH_INSTRUCTION_CACHE_ENABLE();
__HAL_FLASH_DATA_CACHE_ENABLE();
```

Branje podatkov iz Flash-a

- Iz naslovov, ki so shranjeni v Flash pomnilniku beremo enako kot naslove shranjene v RAM

```
uint32_t *p = NASLOV;
uint32_t podatek = *p;
```

Naloga

- Napišite funkcijo, ki prepiše vsebino Flash pomnilnika

  `write_flash(uint32_t size, uint32_t data)`

  - Funkcija naj vedno začne na začetku uporabniškega dela ter pobriše sektorje glede na `size`
- S pomočjo časovnika izmerite čas, ki je potreben za zapis podatkov različnih velikosti:
  - 16KB, 32KB, 64KB, 128KB, 256KB
- Tabelo s časi oddajte kot ločen PDF dokument