Курс по STM32

Лекция #3:

- Ответ на вопрос про начальное тактирование системы.
- Стартовый код микроконтроллера (`entry.S` и `entry.lds`).
- Порты ввода-вывода общего назначения.
- Устранение дребезга кнопки.
- Семисегментный индикатор и динамическая индикация.
- Выдача ДЗ №2.

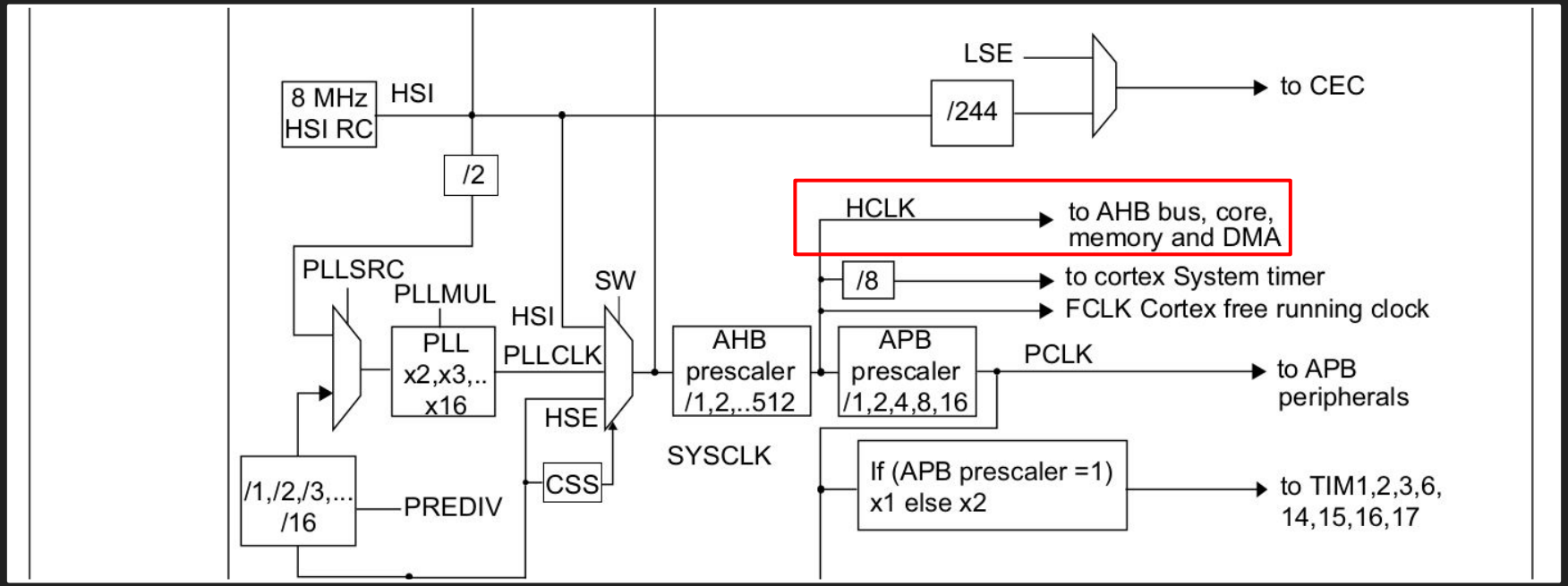
18.02.2023 / 20.02.2023

Ответ на вопрос про начальное тактирование

Начальное тактирование процессора

Из зала: “На какой частоте работает процессор до настройки тактирования?”

1) Где процессор в дереве тактирования?



Начальное тактирование процессора

2) Значение **RCC_CR** по умолчанию: **HSION** = 1 (RC-цепочка 8 Мгц)

6.4.1 Clock control register (RCC_CR)

Address offset: 0x00

Reset value: 0x0000 XX83 where X is undefined.

Access: no wait state, word, half-word and byte access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	PLL RDY	PLLON	Res.	Res.	Res.	Res.	CSS ON	HSE BYP	HSE RDY	HSE ON
						r	rw					rw	rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[7:0]								HSITRIM[4:0]					Res.	HSI RDY	HSION
r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw		r	rw

Начальное тактирование процессора

3) Значение **RCC_CFGR** по умолчанию: (**SW[1:0]** = 0, **HPRE[3:0]** = 0)

6.4.2 Clock configuration register (RCC_CFGR)

Address offset: 0x04

Reset value: 0x0000 0000

HPRE[3:0]: HCLK prescaler

Set and cleared by software to con

0xxx: SYSCLK not divided

1000: SYSCLK divided by 2

1001: SYSCLK divided by 4

Bits 1:0 **SW[1:0]:** System clock switch

Set and cleared by software to select SYSCLK source.

Cleared by hardware to force HSI selection when leaving
of failure of the HSE oscillator used directly or indirectly a
System is enabled).

00: HSI selected as system clock

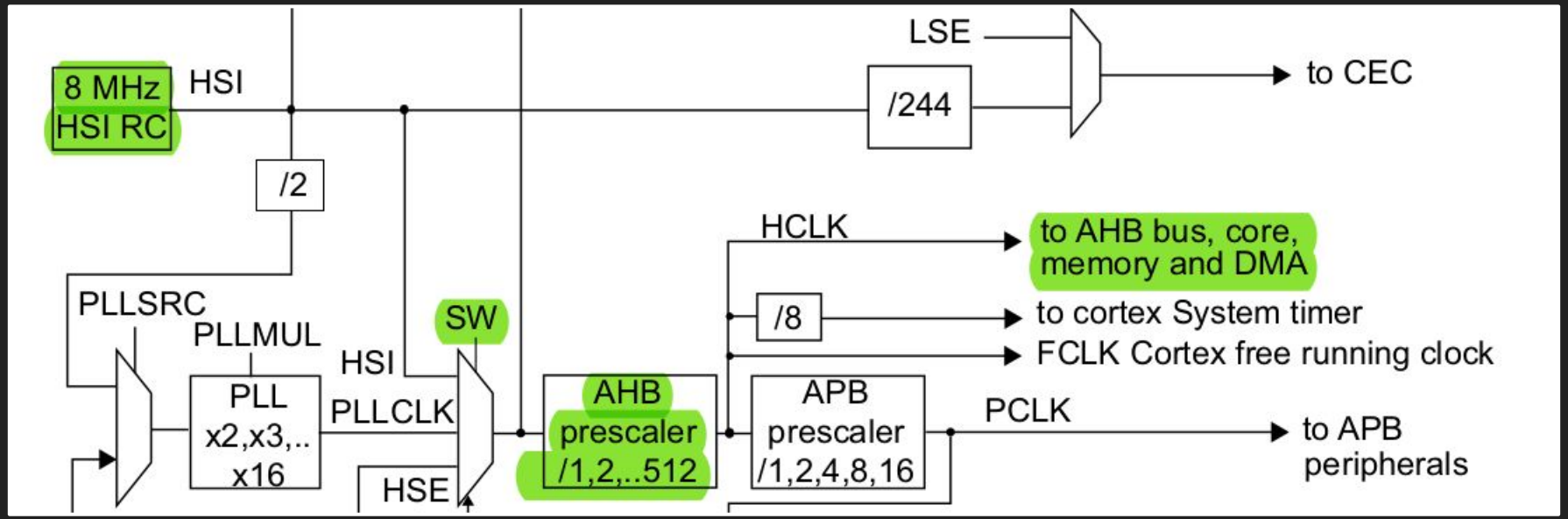
01: HSE selected as system clock

10: PLL selected as system clock

11: HSI48 selected as system clock (when available)

Начальное тактирование процессора

4) Начальная частота процессора – 8 МГц, от RC-цепочки (точность – 1%):



Начальное тактирование SRAM и FLASH

5) По умолчанию, SRAM-память и FLASH-память включены:

Address offset: 0x14

Reset value: 0x0000 0014

Access: no wait state, word, half-word and byte access

Note: When the peripheral clock is not active, the peripheral register values may not be readable by software and the returned value is always 0x0.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSCEN	Res.	IOPF EN	IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	Res.
							rw		rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRC EN	Res.	FLITF EN	Res.	SRAM EN	DMA2 EN	DMA EN
									rw		rw		rw	rw	rw

Стартовый код микроконтроллера

Алгоритм старта процессора

0) Сброс микроконтроллера
1) `stack_pointer   (SP) = *0x00000000`
2) `program_counter (PC) = *0x00000004`
3) Исполнение инструкций по обычной схеме

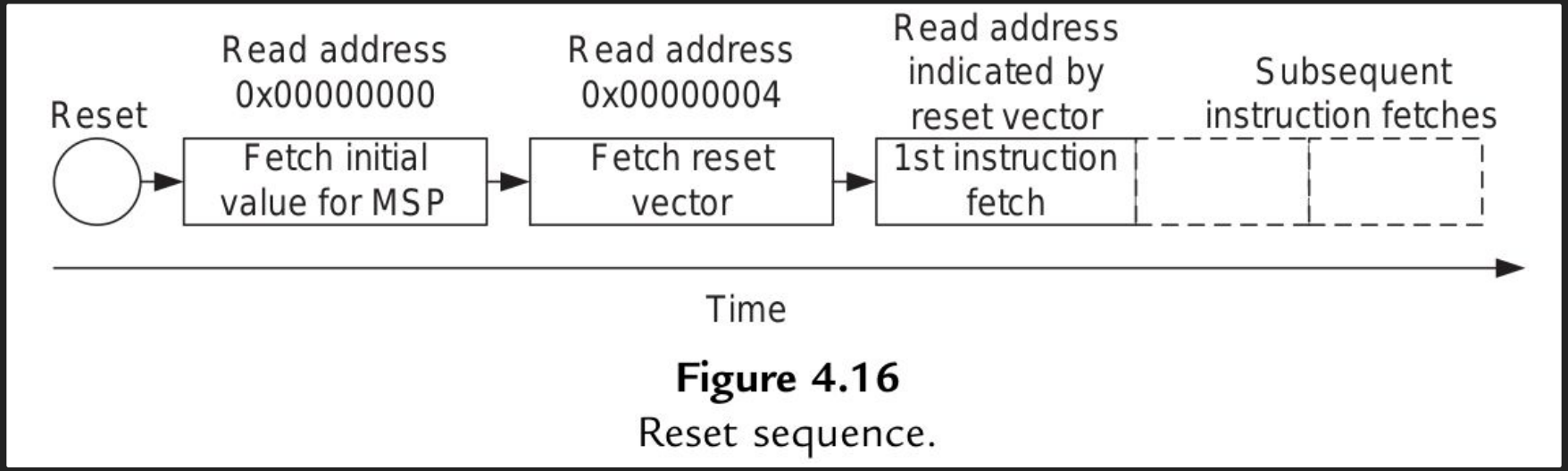


Figure 4.16
Reset sequence.

Инициализация стека и начало исполнения кода

0) Сброс микроконтроллера

1) `SP = *0x00000000 // (*)`
   `SP = 0x20002000`

2) `PC = *0x00000004 // (*)`
   `PC = <reset handler addr>`

3) Исполнение инструкций
по обычной схеме

(*) Из документации на процессор
(docs/cortex_m0_gug.pdf, стр. 20)

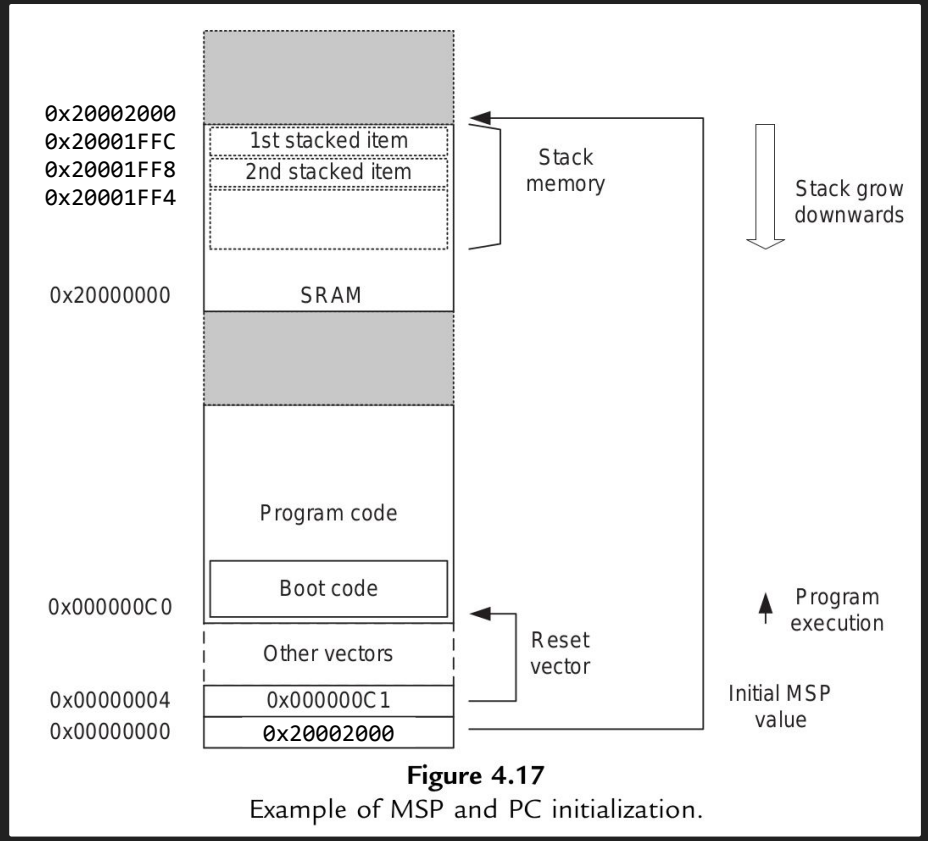


Figure 4.17
Example of MSP and PC initialization.

Устройство стека программы

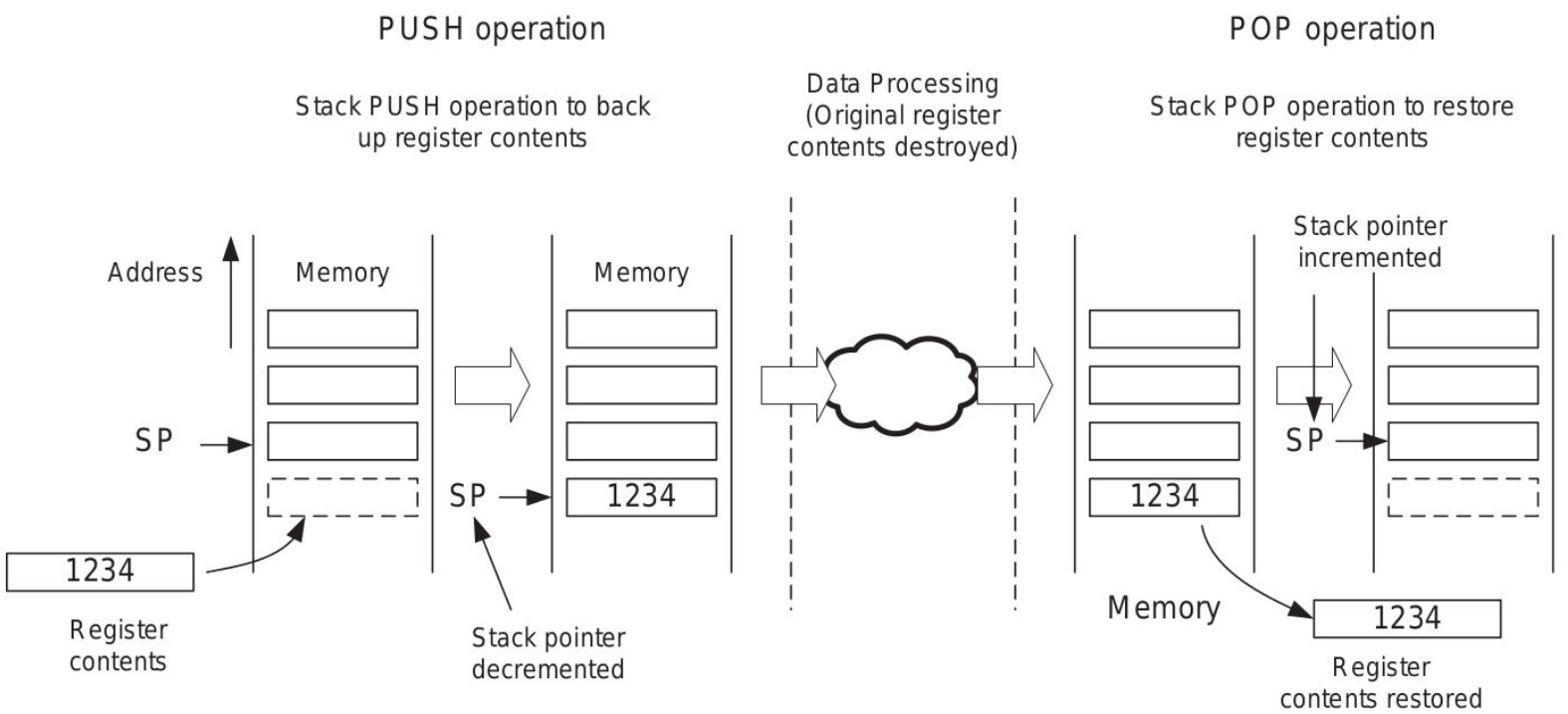


Figure 4.12
Stack PUSH and POP in the Cortex®-M processors.

Стартовый код в `01_blinkled`

```
.syntax unified                 <<<< Выбор набора инструкций (ARM + Thumb)

.section .text                  <<<< Секция .text, для кода

.thumb_func                     <<<< Instruction set – Thumb
.global __reset_handler         <<<< Символ виден в других секциях
__reset_handler:
    blx main                    <<<< Переход в main
__halt:
    b __halt                    <<<< Вечный цикл

.section .vector_table          <<<< Секция .vector_table для Reset Vector
.word __stack_start           // Initial SP
.word __reset_handler         // Reset Handler
```

Инициализация стека и начало исполнения кода

```
~/path/to/stm32f051_rewind/labs/01_blinkled>
❯ arm-none-eabi-objdump -d build/blinkled.elf
...
00000000 <__reset_handler-0x8>:
   0:    20002000     .word    0x20002000 // А откуда взялся SRAM_VADDR?
   4:    00000009     .word    0x00000009 // А почему не 0x00000008?
00000008 <__reset_handler>:
   8:    f000 f88c     bl    124 <main>
0000000c <__halt>:
   c:    e7fe         b.n    c <__halt>
00000010 <board_clocking_init>:
  10:    b580         push    {r7, lr}
...
```

Линкер-скрипт в 01_blinkled

```
ENTRY(__reset_handler);

FLASH_VADDR = 0x08000000;
FLASH_SIZE  = 0x00010000;
SRAM_VADDR  = 0x20000000;
SRAM_SIZE   = 0x00002000;

SECTIONS
{
    . = 0x00000000;

    .text : AT(ADDR(.text) + FLASH_VADDR)
    {
        KEEP(*(.vector_table));
        *(.text)
        *(.rodata)
    }

    __stack_start = SRAM_VADDR + SRAM_SIZE;
}
```

<<< Точка входа в программу

<<< Параметры FLASH-памяти

<<< Параметры SRAM-памяти

<<< “.” – текущий виртуальный адрес

<<< Секция `.text` располагается по адресу FLASH-памяти

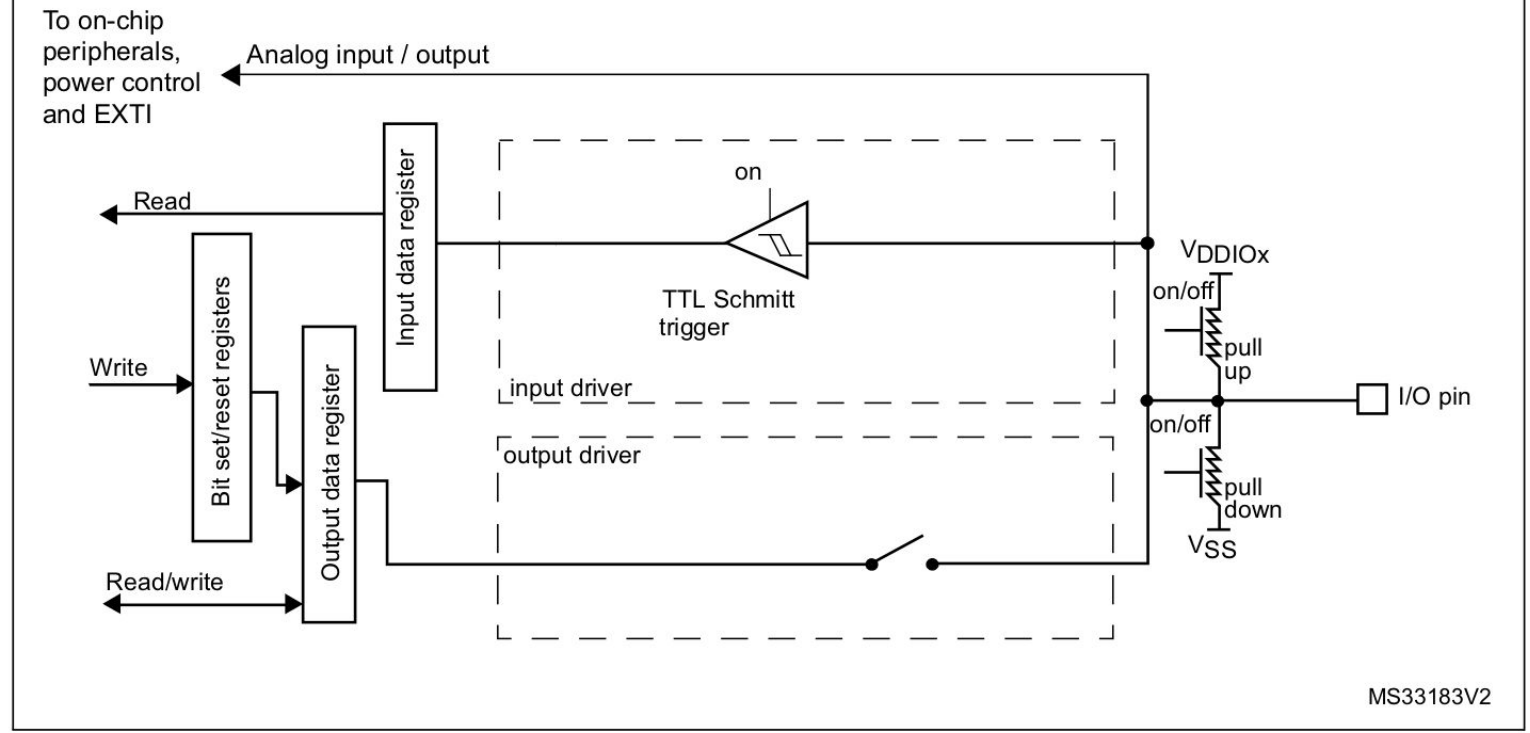
<<< Список входных секций, размещаемых в секции `.text`

<<< Стек размещается в SRAM-памяти

Порты ввода-вывода общего назначения (GPIO)

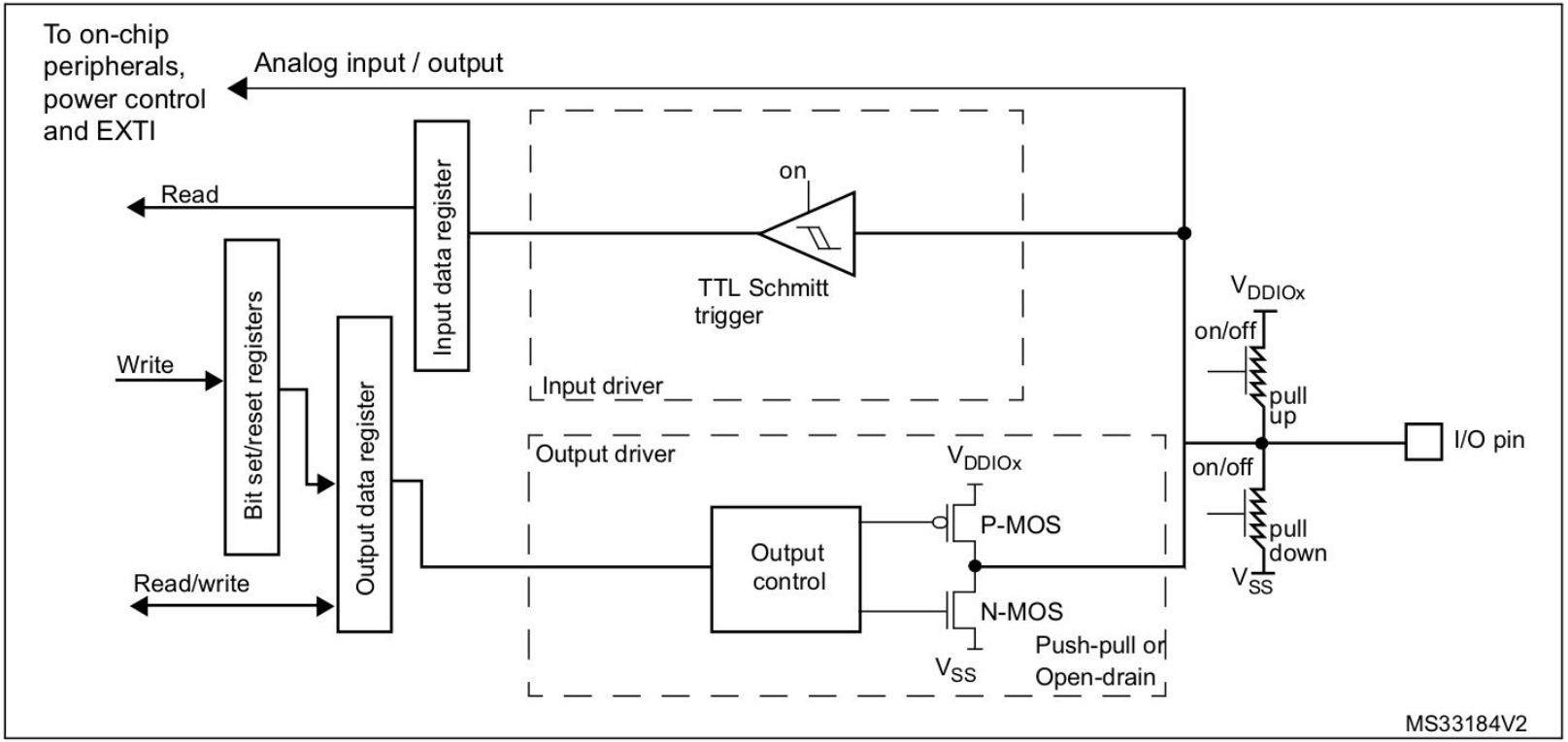
Пин GPIO: цифровой вход

Figure 17. Input floating/pull up/pull down configurations



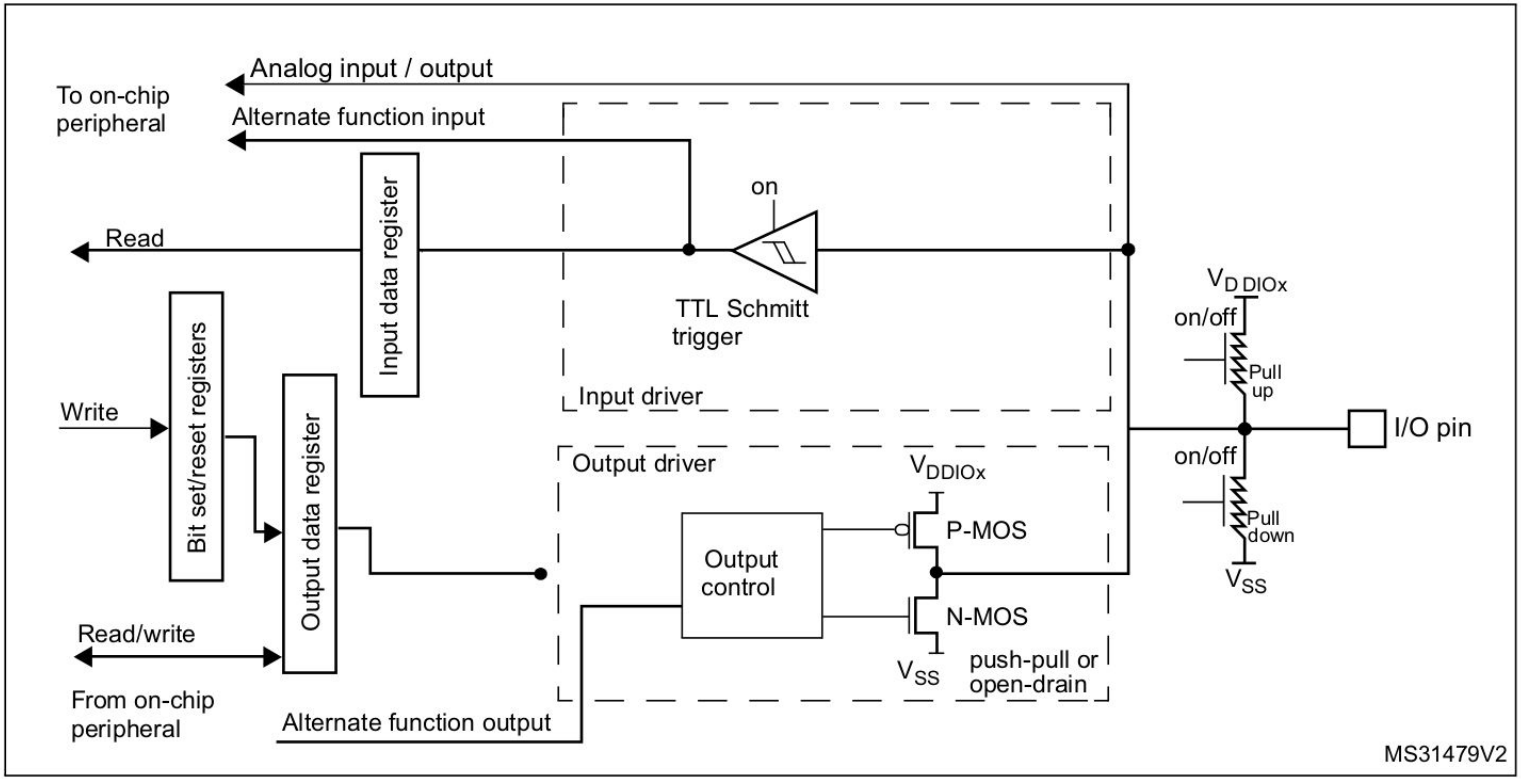
Пин GPIO: цифровой выход

Figure 18. Output configuration



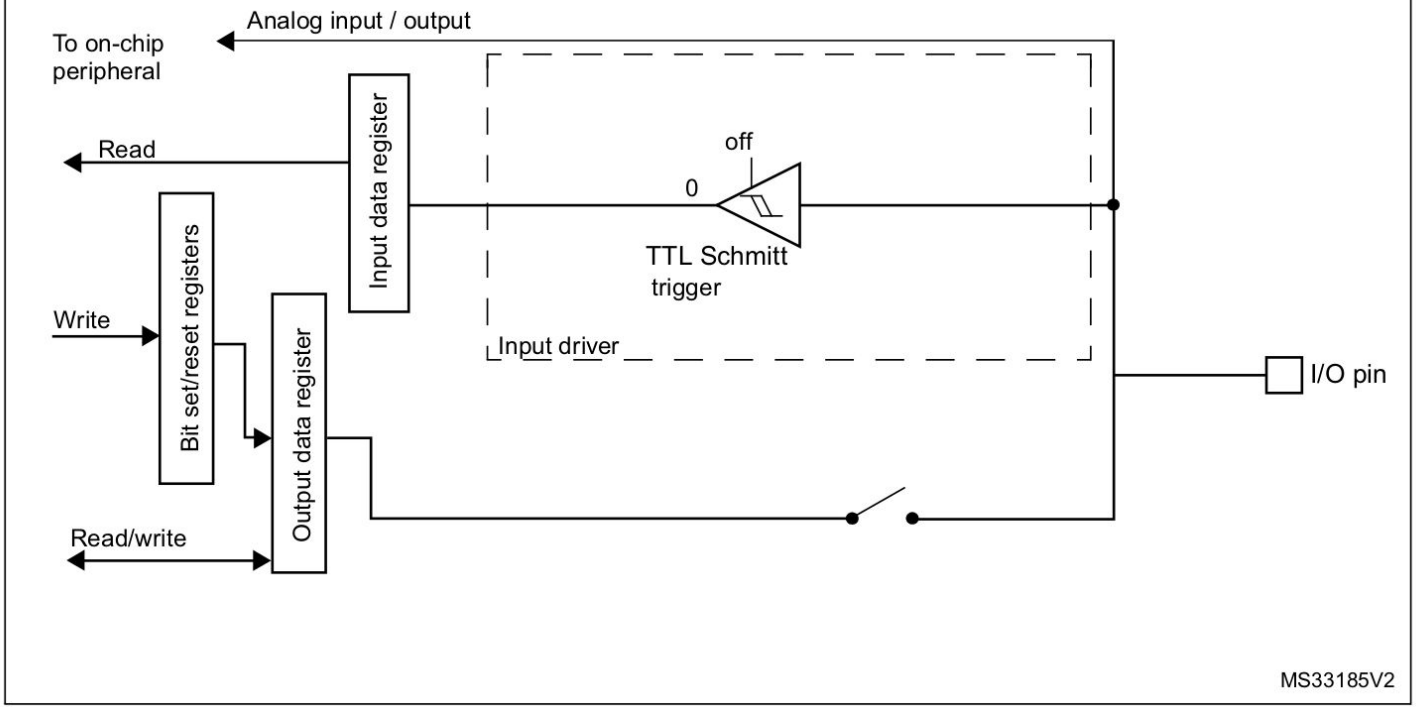
Пин GPIO: альтернативный режим

Figure 19. Alternate function configuration



Пин GPIO: аналоговый режим

Figure 20. High impedance-analog configuration



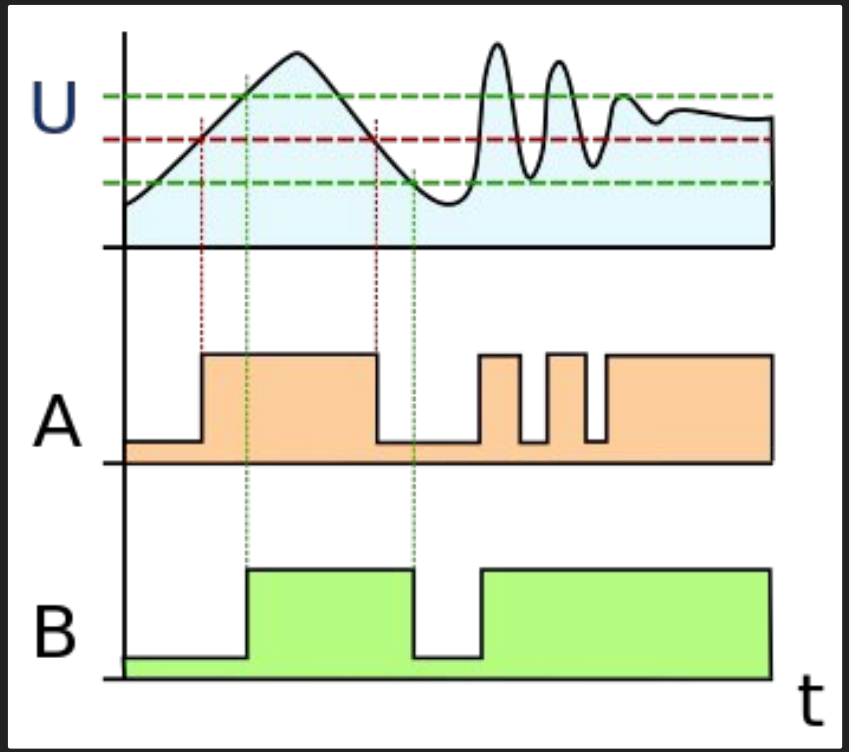
Пин GPIO: подтягивающие резисторы

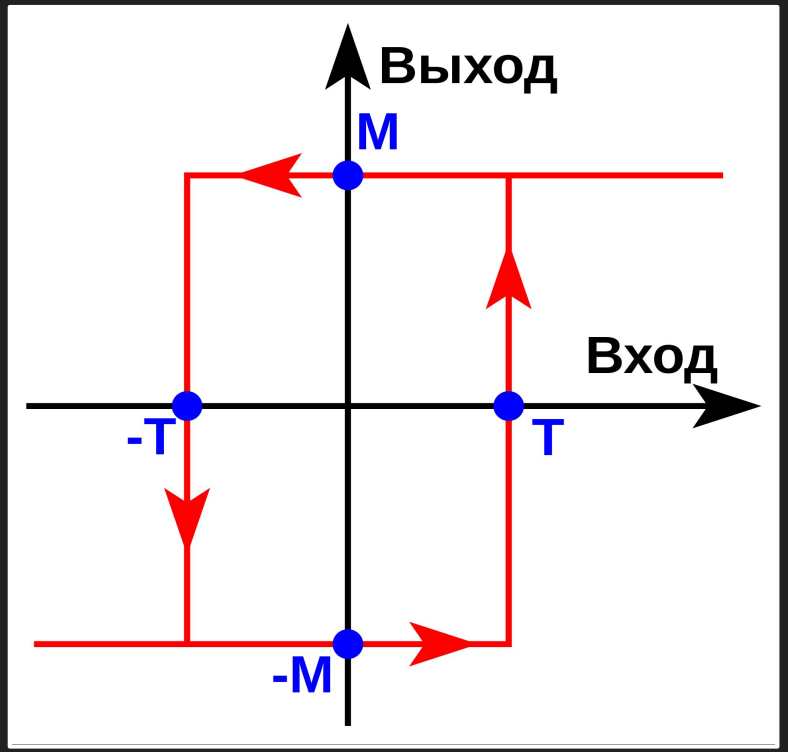
Тип подтяжки	На входе “0”	На входе “1”	High-Z (вход разомкнут)
None	**0**	**1**	**Шум, наводки питания**
Pull-up	**0**	**1**	**1**
Pull-down	**0**	**1**	**0**

Применения:

- Обработка ввода с кнопок.
- Протоколы передачи: установка значение на шине, когда передача не ведётся или передатчик не подключён.

Пин GPIO: триггер Шмитта





Пин GPIO: режимы цифрового вывода

Режим вывода	В регистре “0”	В регистре “1”
Push-pull	**0**	**1**
Open-Drain	**0**	**High-Z**

Применения:

- Протоколы передачи: шины с N приёмопередатчиками.
- Связь 3.3В и 5В логики.

Пин GPIO: регистр GPIO_BSRR

Зачем создавать ещё один регистр, если есть GPIO_ODR?

Bits 31:16 **BRy:** Port x reset bit y (y = 0..15)

These bits are write-only. A read to these bits returns the value 0x0000.

0: No action on the corresponding ODRx bit

1: Resets the corresponding ODRx bit

Note: If both BSx and BRx are set, BSx has priority.

Bits 15:0 **BSy:** Port x set bit y (y= 0..15)

These bits are write-only. A read to these bits returns the value 0x0000.

0: No action on the corresponding ODRx bit

1: Sets the corresponding ODRx bit

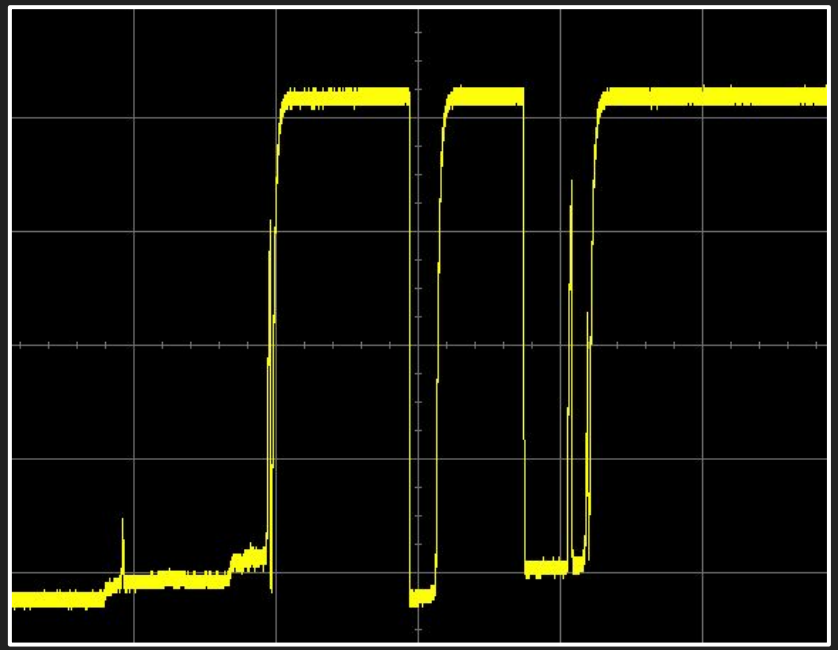
Цифровой выход: регистр GPIO_BSRR

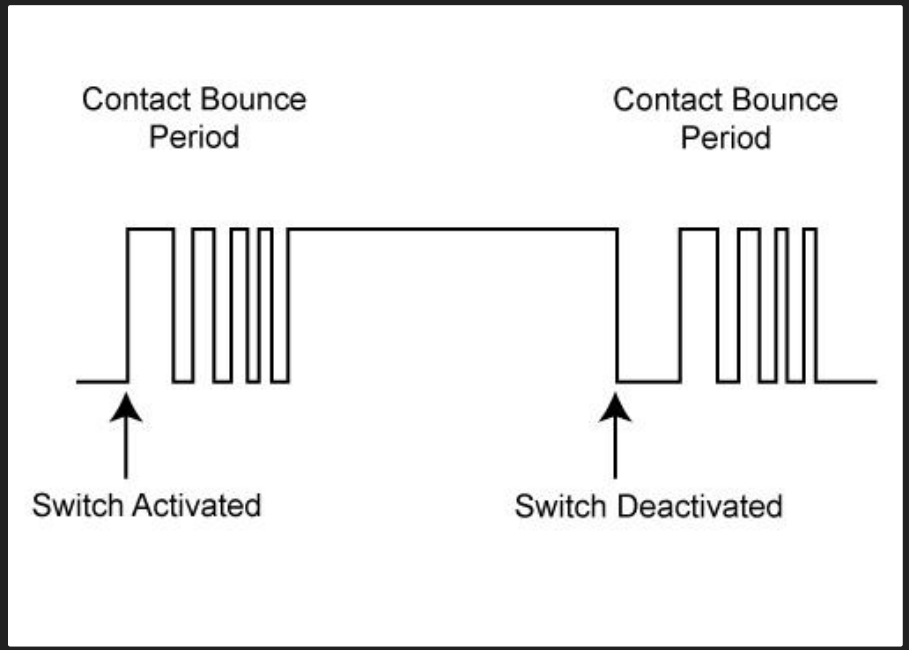
```
*(volatile uint32_t*)(uintptr_t)0x48000814U |=  0x100U;
```

```
~/path/to/stm32f051_rewind/labs/01_blinkled>
❯ arm-none-eabi-objdump -d build/blinkled.elf
...
 130:    4b08    ldr     r3, [pc, #32]    // r3 = 0x48000814;
 132:    681a    ldr     r2, [r3, #0]     // r2 = *r3
 134:    4b07    ldr     r3, [pc, #28]    // r3 = 0x48000814;
 136:    2180    movs    r1, #128         // r1 = 0x80;
 138:    0049    lsls    r1, r1, #1       // r1 = (r1 << 1);
 13a:    430a    orrs    r2, r1           // r2 |= r1;
 13c:    601a    str     r2, [r3, #0]     // *r3 = r2;
...
```

Дребезг контактов

Дребезг контактов: о проблеме





Дребезг контактов: решения

Аппаратные способы:

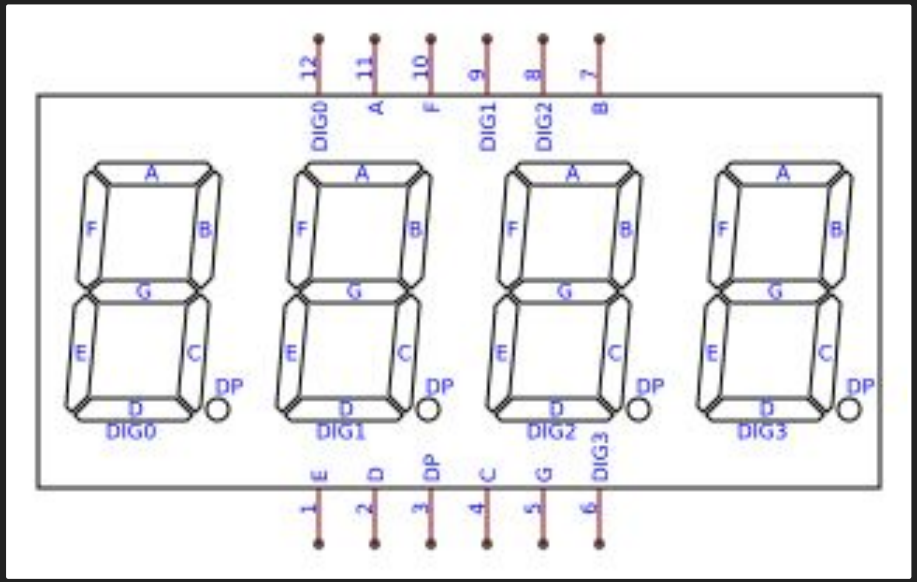
- Фильтр Нижних Частот (RC-цепочка или более сложный фильтр)
- Цифровая схема, реализующая ФНЧ

Программные способы:

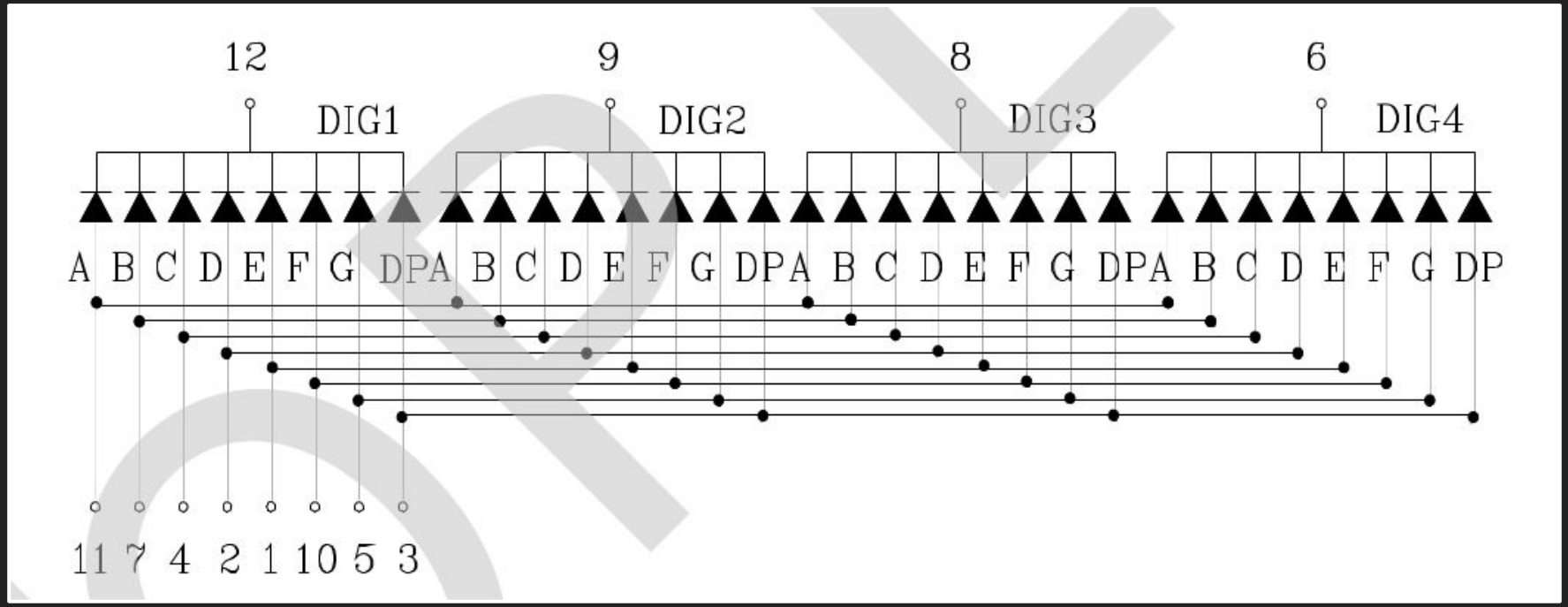
- “Кнопка нажата 10 мс подряд => срабатывание”
- “Схема с гистерезисом, триггер Шмитта”

Семисегментный индикатор

Семисегментный индикатор: устройство



Динамическая индикация



ДЗ №2: 02_gpio

Требования к ДЗ №2

```
[1] Отрефакторить код 02_gpio и разметить все регистры:
    [ ] Регистры используются только по их именам.
    [ ] Используются биты регистров только по их именам.
[2] Реализовать игру "пальчики".
    [ ] Две кнопки-пальца.
    [ ] Два диода – статусы игроков.
    [ ] Семисегментный индикатор – счёт за партию.
    Полный список требований -- в README.md
[3] Реализовать обработку дребезга кнопки "с гистерезисом".

[2’] Альтернативный вариант -- появится на сл.лекции!
```

Спасибо за внимание!

Красивые иконки взяты с сайта handdrawngoods.com