STM32 course

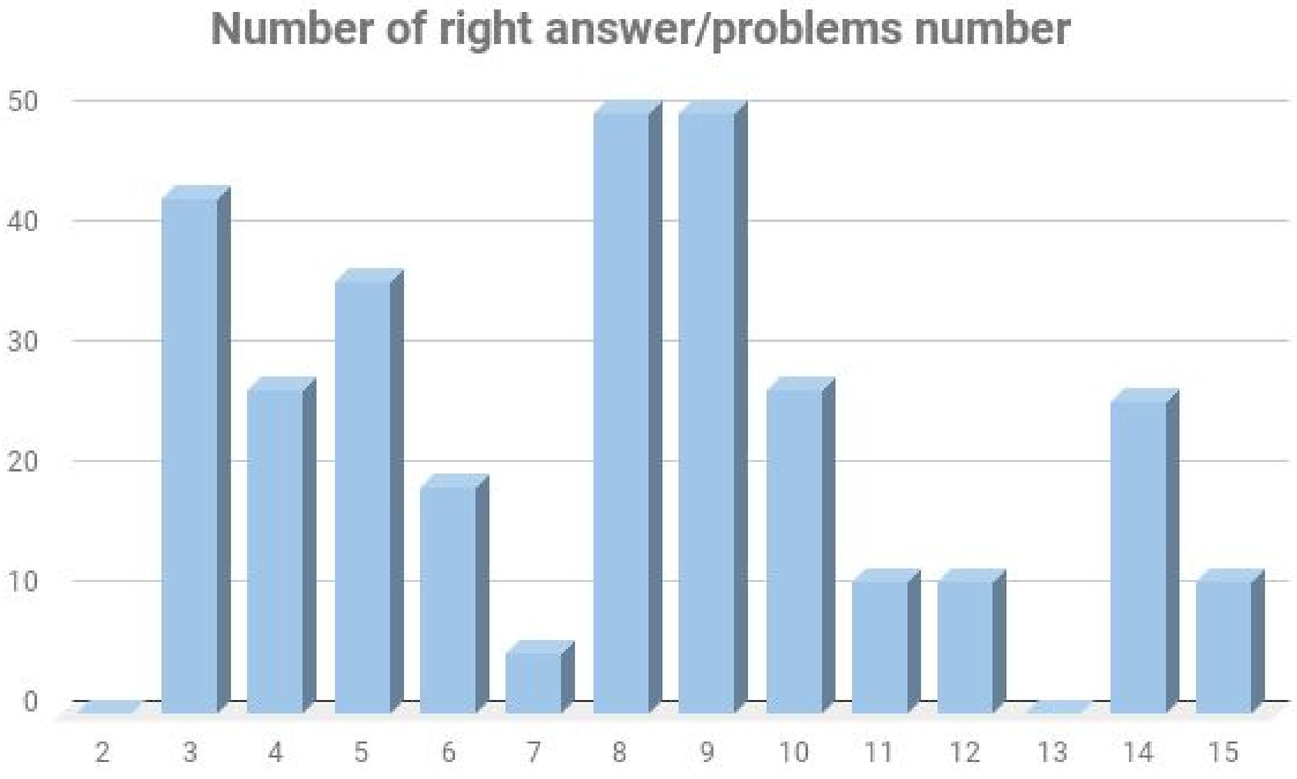
Getting started with ARM Cortex-M0

Moscow Institute of Physics and Technology

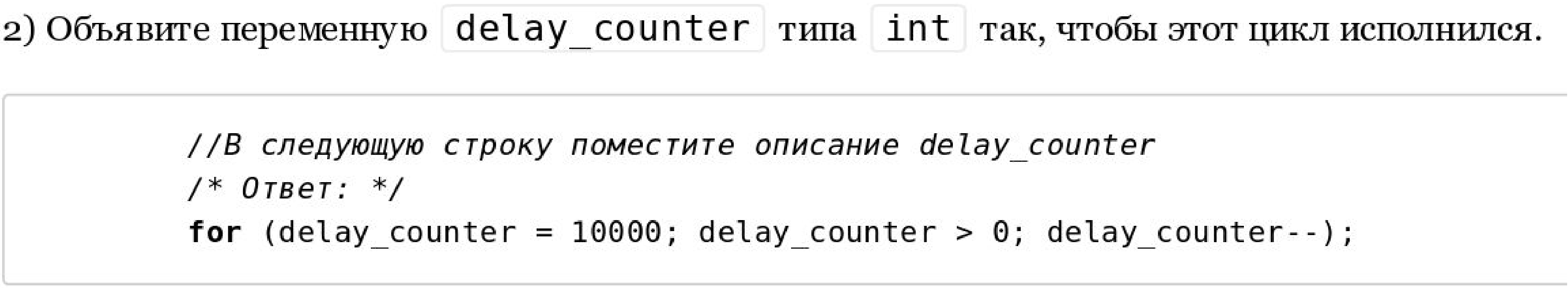
Outline

- Test results
- ARM Cortex-M0
  - Core
  - Interrupts
  - Debugging
  - NVIC
  - Bus
- Software development toolchain
  - Tools
  - Ready-to-use makefile, docs
- Practice

Test results



Test results. The hardest task ever



2) Объявите переменную `delay_counter` типа `int` так, чтобы этот цикл исполнился.

```
        //В следующую строку поместите описание delay_counter
        /* Ответ: */
        for (delay_counter = 10000; delay_counter > 0; delay_counter--);
```

Right answer:

volatile int delay_counter;

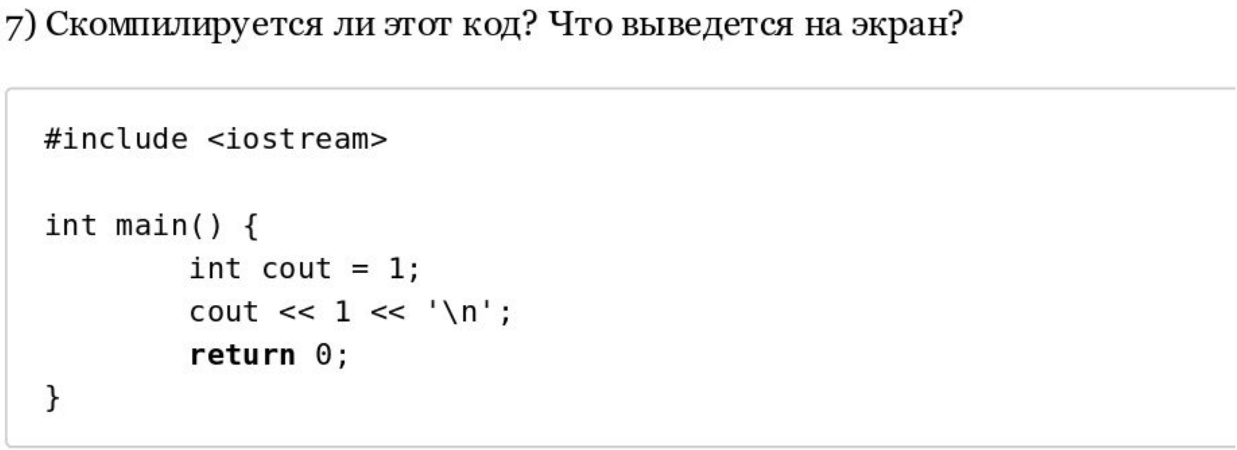
Test results

6) Чему равно максимальное значение переменной типа int8_t?

Right answer:

127

Test results



7) Скомпилируется ли этот код? Что выведется на экран?

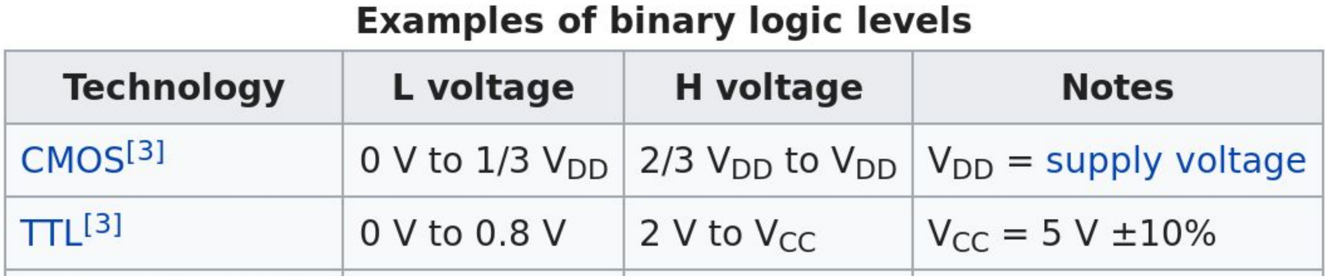
```
#include <iostream>

int main() {
        int cout = 1;
        cout << 1 << '\n';
        return 0;
}
```

Right answer:

Sure. Nothing.

Test results

10) Стандартное(-ые) напряжение(-я) логической единицы?



Examples of binary logic levels

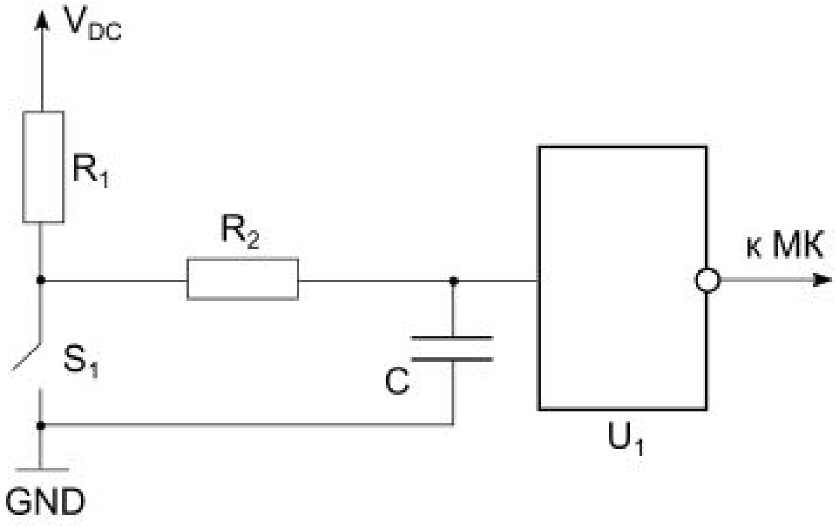
Technology	L voltage	H voltage	Notes
CMOS[3]	0 V to 1/3 V_{DD}	2/3 V_{DD} to V_{DD}	V_{DD} = supply voltage
TTL[3]	0 V to 0.8 V	2 V to V_{CC}	V_{CC} = 5 V ±10%

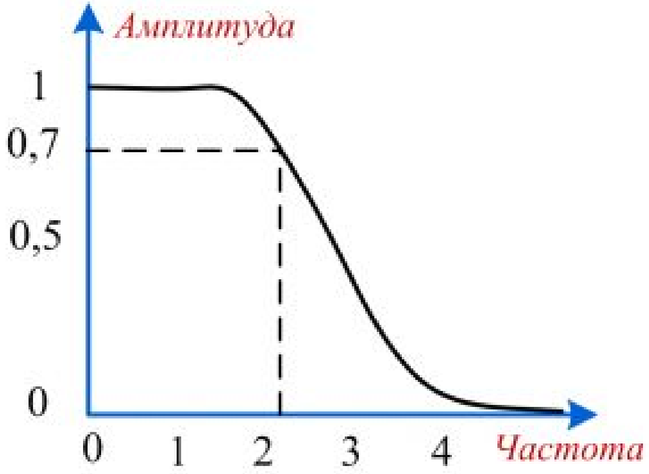
11) С помощью какого устройства можно 3 пинами управлять 8 светодиодами?

Мультиплексор (multiplexer), сдвиговый регистр (shift register), дешифратор (decoder)

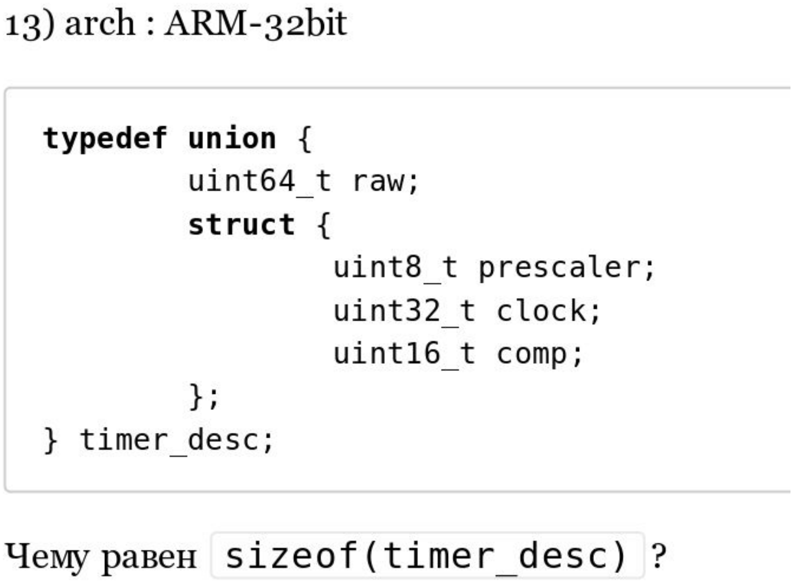
Test results

12) С помощью чего можно защититься от дребезга кнопки аппаратно?





Test results



13) arch : ARM-32bit

```
typedef union {
        uint64_t raw;
        struct {
                uint8_t prescaler;
                uint32_t clock;
                uint16_t comp;
        };
} timer_desc;
```

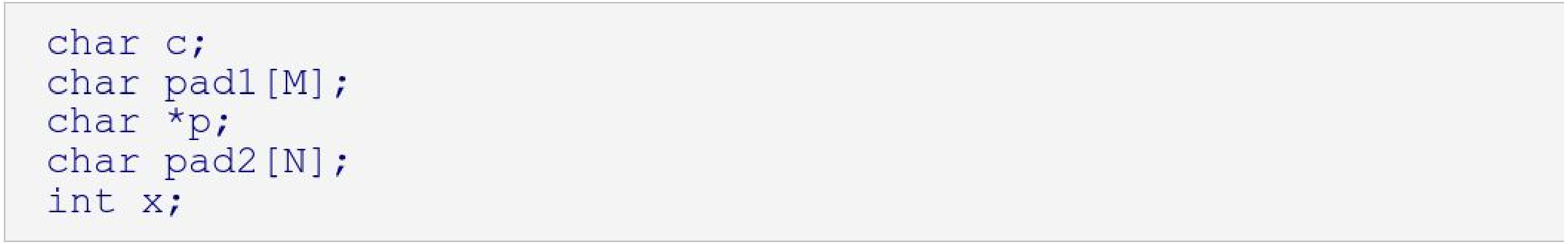
Чему равен `sizeof(timer_desc)` ?

Right answer:

Compiler specific
arm-none-eabi-gcc(LE) gives 16

ALSO, on x86-64: 16

A few words about struct alignment



```
char c;
char pad1[M];
char *p;
char pad2[N];
int x;
```

N = 0 (anyway)

BE: M = 0

LE: M = 3 (32bit arch), 7(64bit arch)

For more details -> http://www.catb.org/esr/structure-packing/

Test results

14) С какой частотой мигает в бытовой электросети:

а) лампа накаливания -> **0 Hz**

б) неоновая лампа -> **100 Hz**

в) светодиод -> **50 Hz**

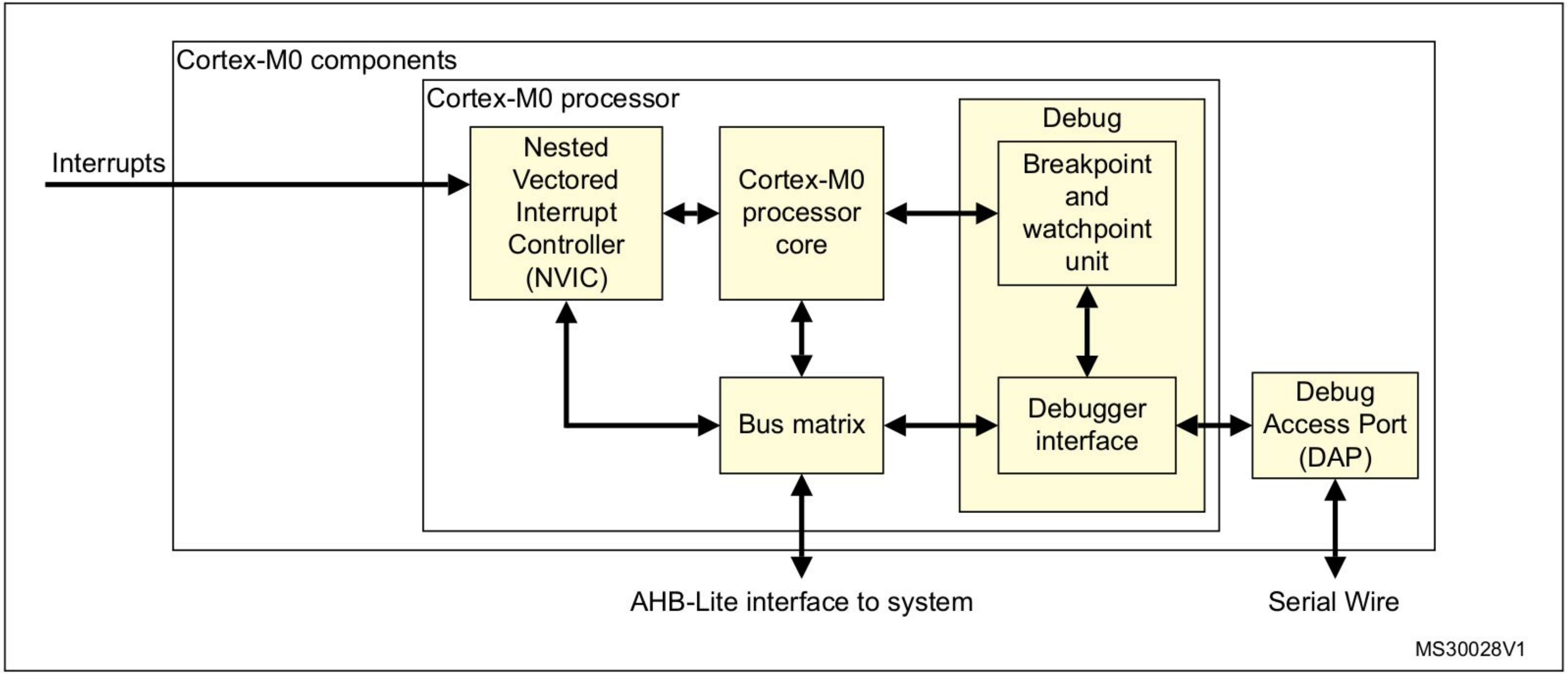
15) Напишите строчку, чтобы проинвертировать 4-й бит в 32-битной беззнаковой переменной mask

```
mask ^= (1U << 4);
mask = (mask & ~(1U << 4)) | (~mask & (1U << 4))
```

ARM Cortex-M0

- a simple architecture that is easy to learn and program
- ultra-low power, energy efficient operation
- excellent code density
- deterministic, high-performance interrupt handling
- upward compatibility with Cortex-M processor family.

ARM Cortex-M0 (ARMv6-M)



ARM Cortex-M0. Processor modes

Thread mode

Used to execute application software.

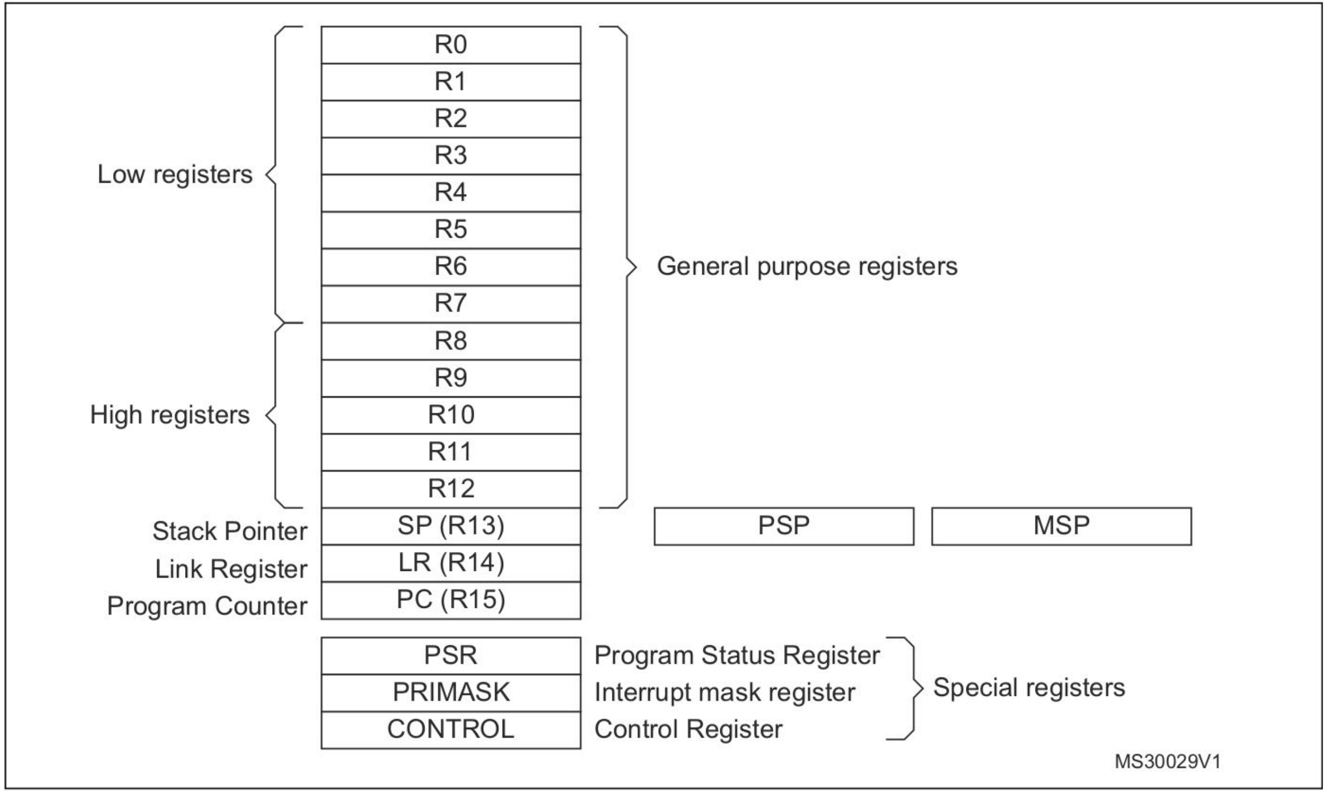
The processor enters Thread mode when it comes out of reset.

Handler mode

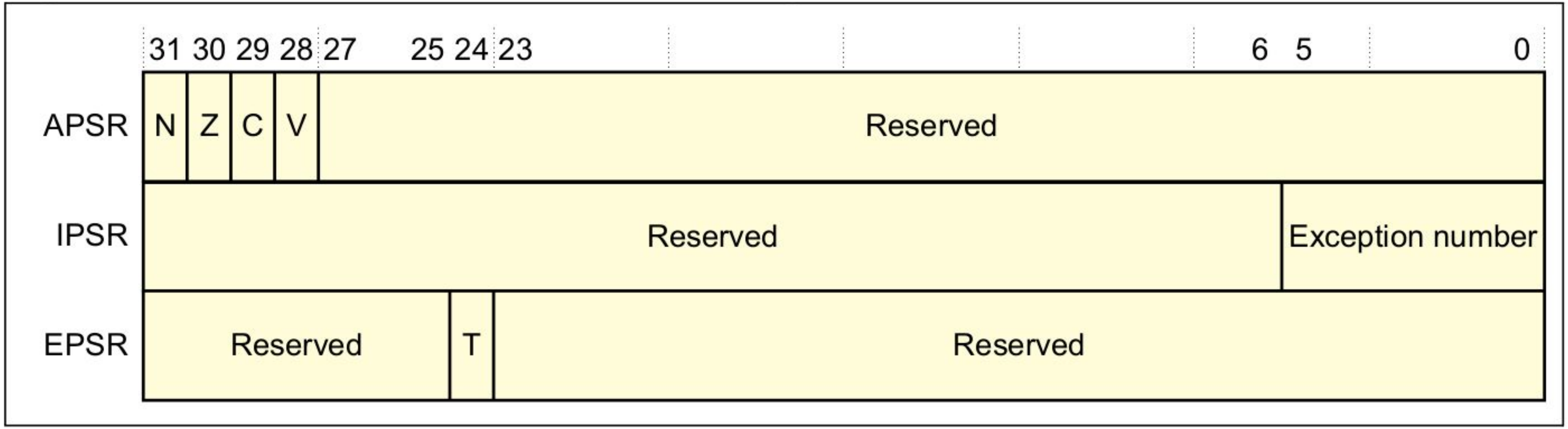
Used to handle exceptions.

The processor returns to Thread mode when it has finished exception processing.

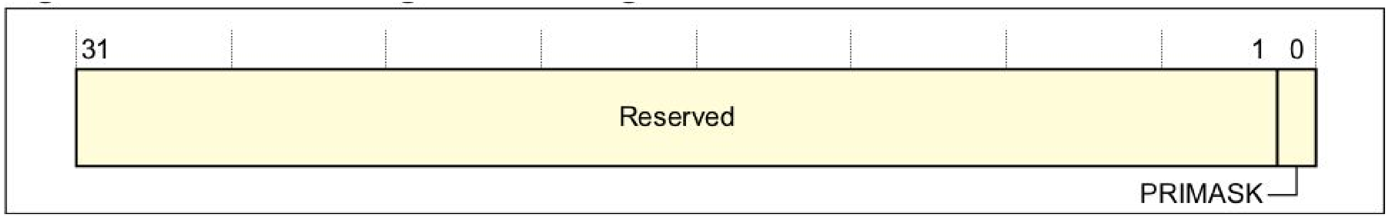
ARM Cortex-M0. Processor core registers

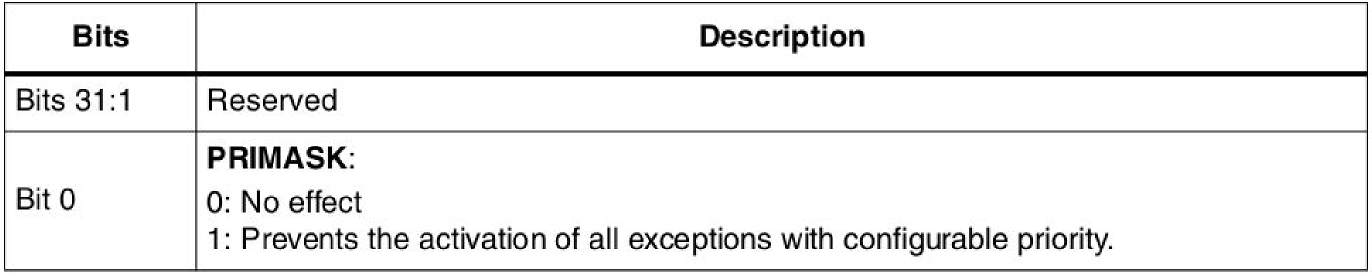


ARM Cortex-M0. Program Status Register (PSR)



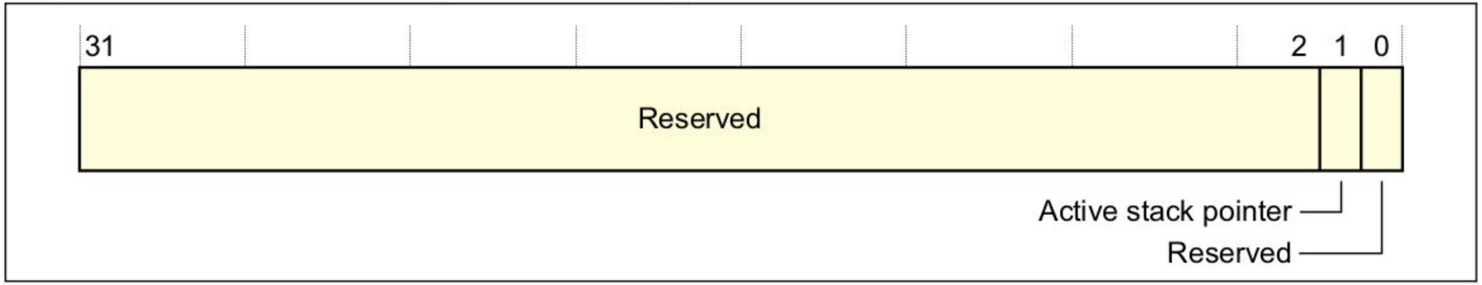
ARM Cortex-M0. PRIMASK

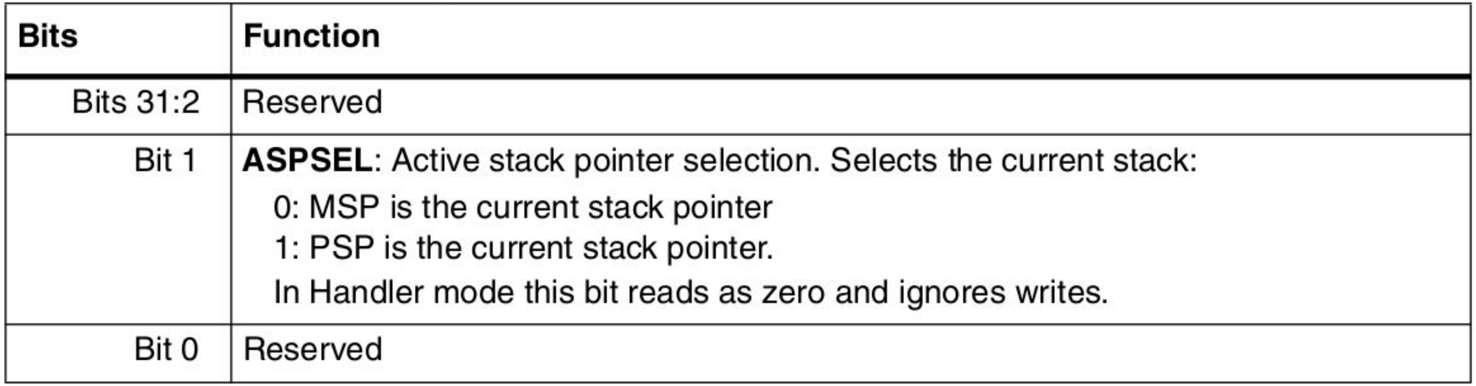




Bits	Description
Bits 31:1	Reserved
Bit 0	**PRIMASK**: 0: No effect 1: Prevents the activation of all exceptions with configurable priority.

ARM Cortex-M0. CONTROL





Bits	Function
Bits 31:2	Reserved
Bit 1	**ASPSEL**: Active stack pointer selection. Selects the current stack: 0: MSP is the current stack pointer 1: PSP is the current stack pointer. In Handler mode this bit reads as zero and ignores writes.
Bit 0	Reserved

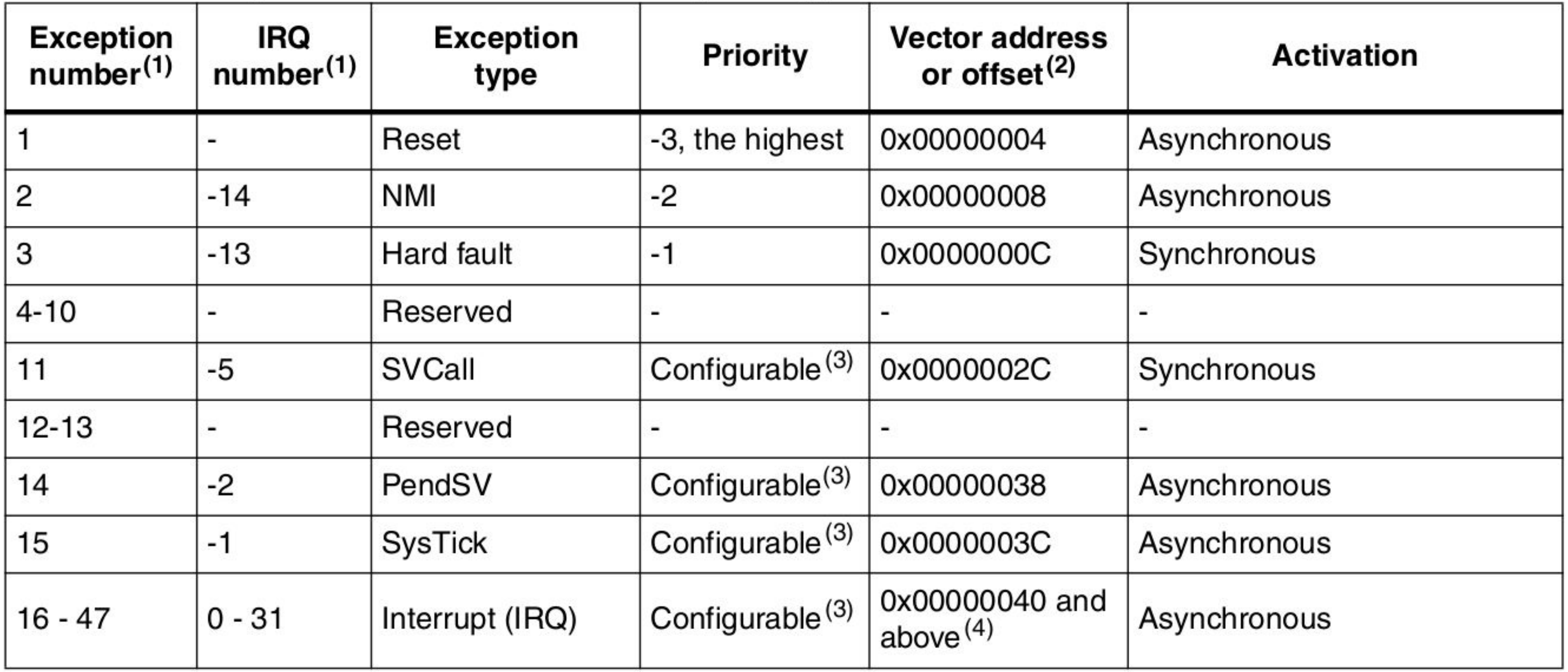
ARM Cortex-M0. Exceptions and interrupts

In terms of ARM Cortex-M0 core - there is no difference between exceptions and interrupts

- 64 entries in vector table
- The first entry is **stack address** (End of RAM)
- System interrupts:
  - Reset, NMI, Hard Fault, SVCall, PendSV, SysTick
- Up to 32 prioritizable peripheral interrupts

It is time to check it out!

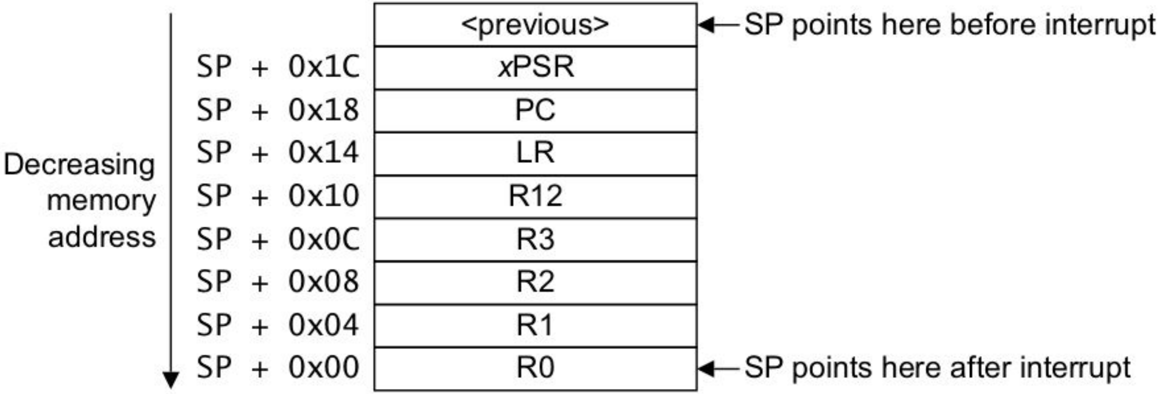
ARM Cortex-M0. Exceptions and interrupts



Exception number[1]	IRQ number[1]	Exception type	Priority	Vector address or offset[2]	Activation
1	-	Reset	-3, the highest	0x00000004	Asynchronous
2	-14	NMI	-2	0x00000008	Asynchronous
3	-13	Hard fault	-1	0x0000000C	Synchronous
4-10	-	Reserved	-	-	-
11	-5	SVCall	Configurable[3]	0x0000002C	Synchronous
12-13	-	Reserved	-	-	-
14	-2	PendSV	Configurable[3]	0x00000038	Asynchronous
15	-1	SysTick	Configurable[3]	0x0000003C	Asynchronous
16 - 47	0 - 31	Interrupt (IRQ)	Configurable[3]	0x00000040 and above[4]	Asynchronous

ARM Cortex-M0. Exceptions and interrupts

- Tail-chaining
- Late-arriving
- Return
- Preemption



Decreasing memory address

	<previous>
SP + 0x1C	*x*PSR
SP + 0x18	PC
SP + 0x14	LR
SP + 0x10	R12
SP + 0x0C	R3
SP + 0x08	R2
SP + 0x04	R1
SP + 0x00	R0

SP points here before interrupt

SP points here after interrupt

ARM Cortex-M0. Memory map

Code	0x00000000 - 0x1FFFFFFF
SRAM	0x20000000 - 0x3FFFFFFF
Peripheral	0x40000000 - 0x5FFFFFFF
External RAM	0x60000000 - 0x9FFFFFFF
External device	0xA0000000 - 0xDFFFFFFF
Private peripheral bus	0xE0000000 - 0xE00FFFFF
Device	0xE0100000 - 0xFFFFFFFF

Nested vectored interrupt controller (NVIC)

- void NVIC_EnableIRQ(IRQn_Type IRQn)
- void NVIC_DisableIRQ(IRQn_Type IRQn)
- void NVIC_SetPriority(IRQn_Type IRQn, uint32_t priority)
- uint32_t NVIC_GetPriority(IRQn_Type IRQn)

Software development toolchain

- gcc-arm-none-eabi
- vim
- gdb-arm-none-eabi
- openocd
- st-flash, st-util

Template project link:
https://github.com/edosedgar/stm32f0_ARM/tree/master/TemplateProject

Documentation link:
https://github.com/edosedgar/stm32f0_ARM/tree/master/doc

Practice

It is time to start blinking a LED