

Программирование микроконтроллеров STM32

*Контроллер прерываний. Системные
прерывания. Периферийные прерывания*

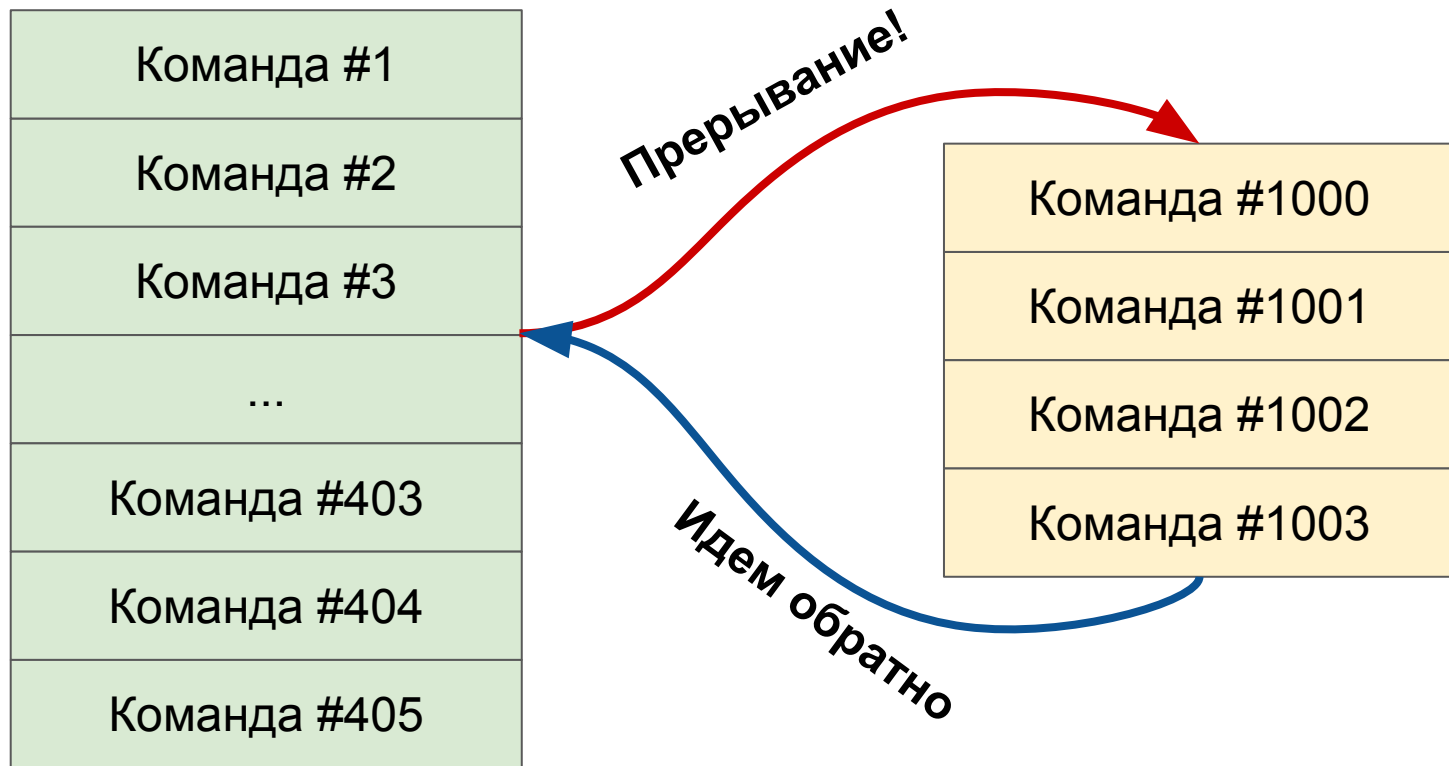
Как обработать внешнее событие?



www.Vecto.rs · 22705

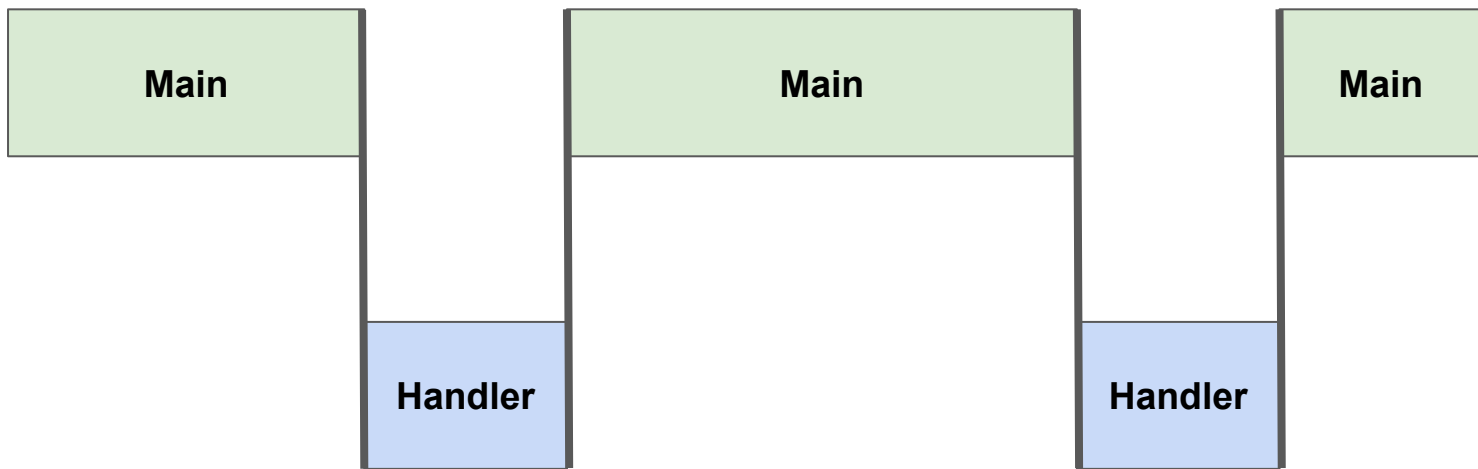
- Постоянный опрос
- Прерывание

Действия при прерывании

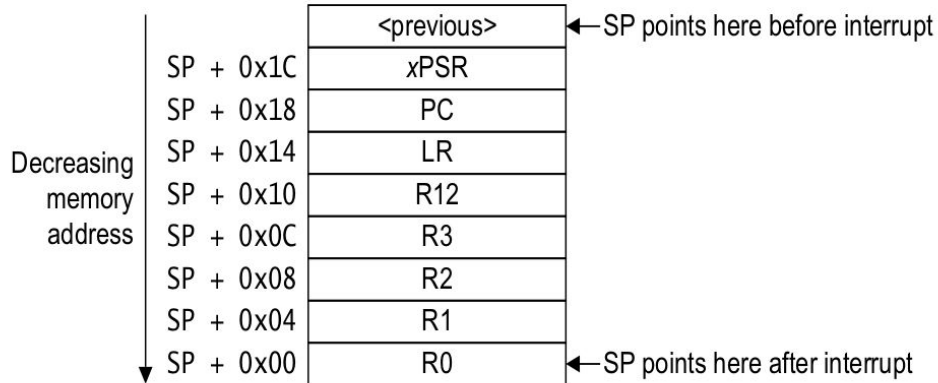
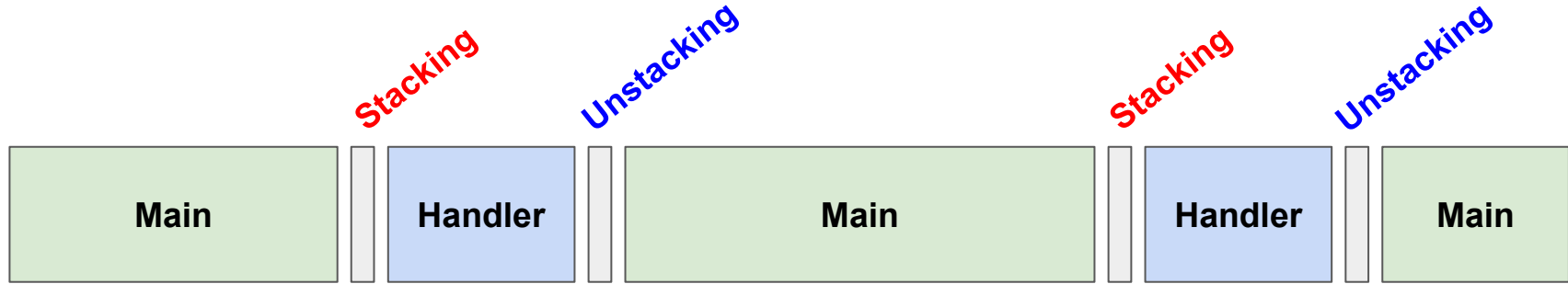


*прерывания не всегда “между командами”

Время

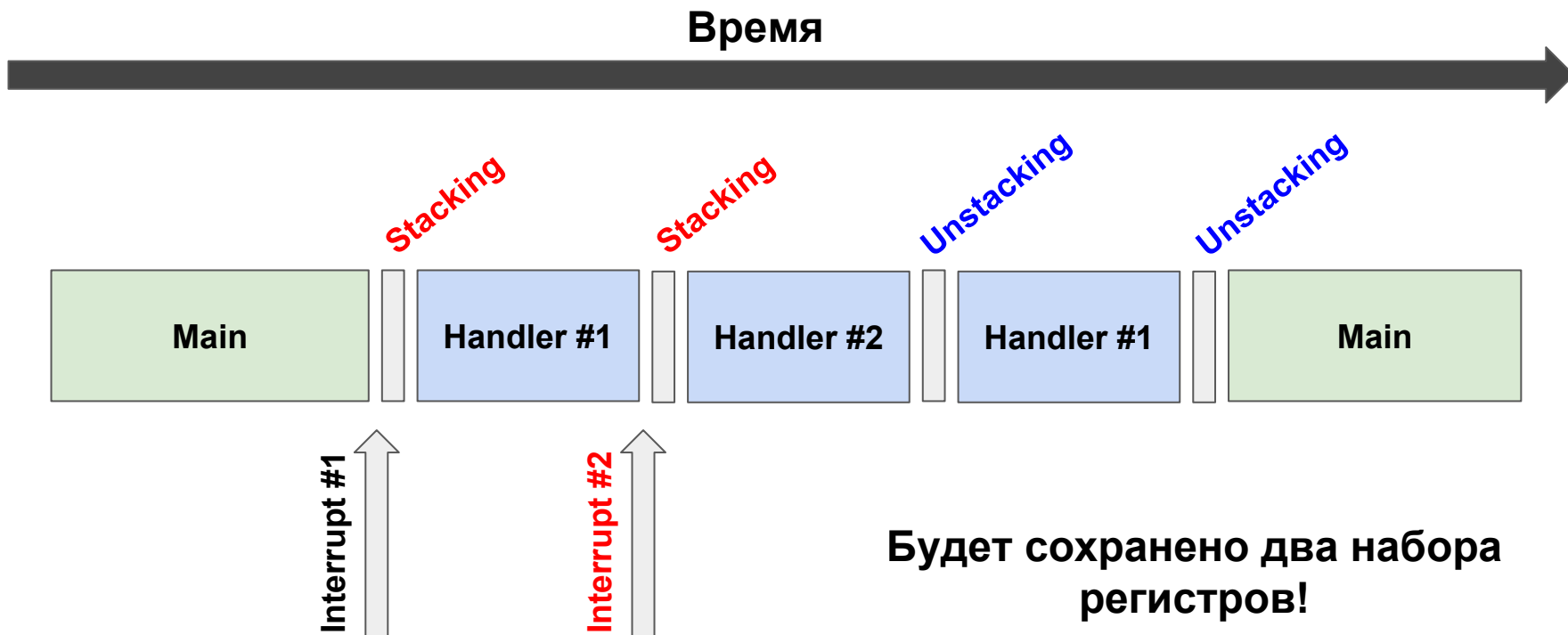


Время

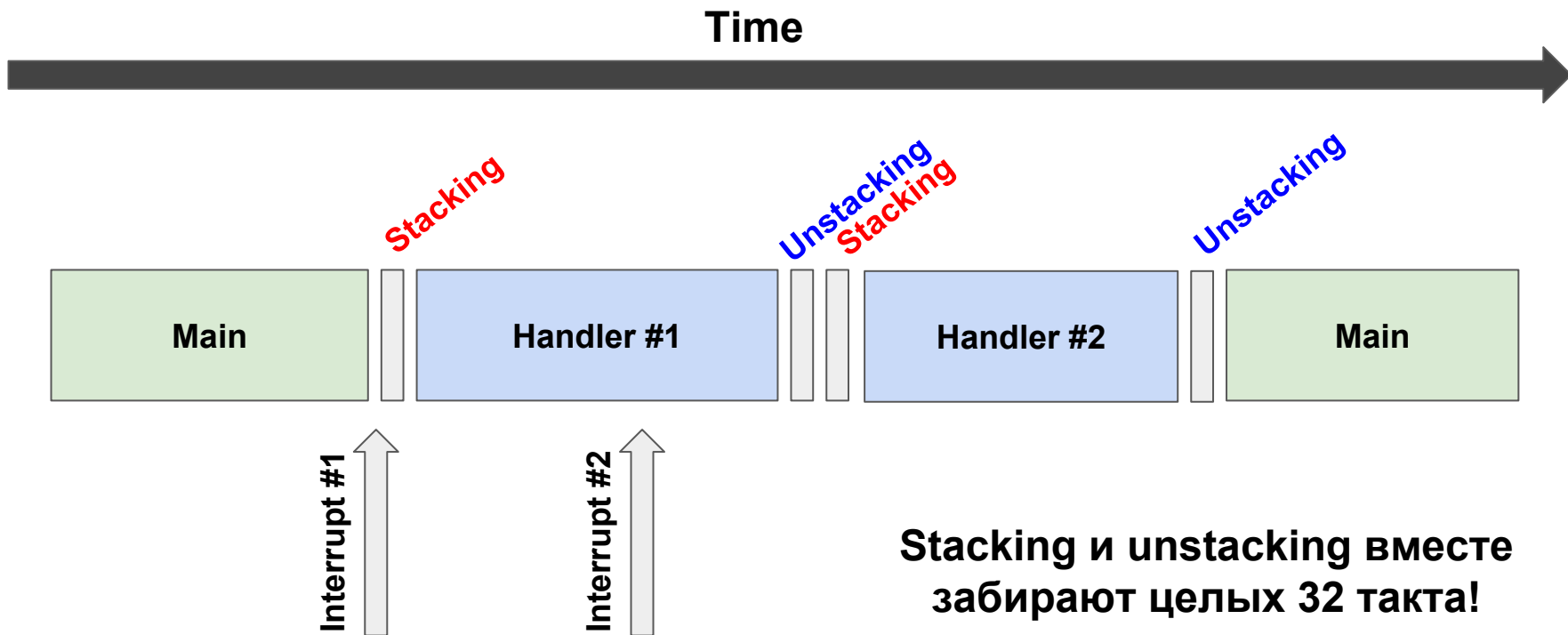


**Each
stacking/unstacking
procedure takes 16
cycles and atomic!**

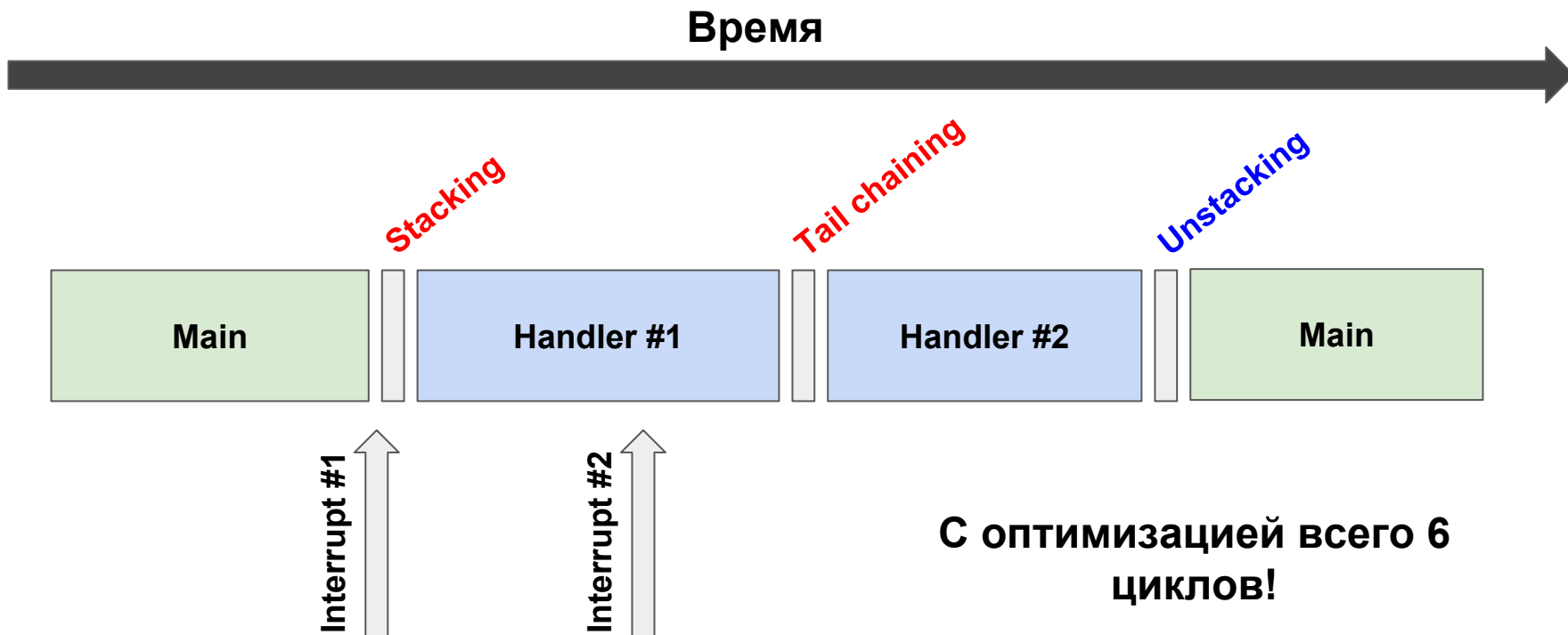
Вложенные прерывания. Вытеснение



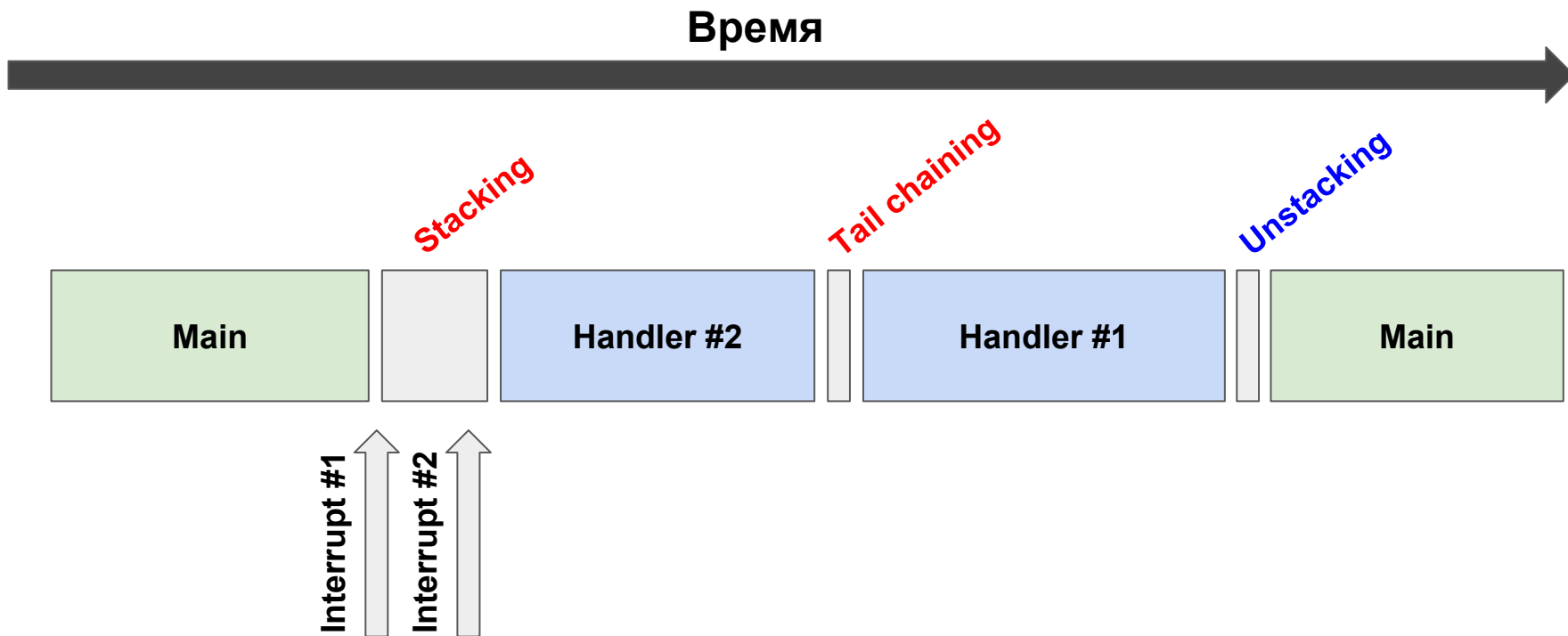
Вложенные прерывания. Ожидание



Вложенные прерывания. Tail-chaining



Вложенные прерывания. Late-arriving



Контроллер прерываний NVIC

- Быстрая обработка входящих прерываний (16 тактов)
- Набор оптимизаций при обработке нескольких прерываний
- Отсутствие джиттера в силу поддержки прерываний мультицикловых команд (STR, LDR)
- 4 возможных уровня приоритета
- 3 статуса прерывания (неактивный, ожидающий, активный), но также активный и ожидающий одновременно

**Exception number =
(IRQ number + 16) * 4**

Exception number	IRQ number	Vector	Offset
47	31	IRQ31	0xBC
.		.	.
.		.	.
.		.	.
18	2	IRQ2	0x48
17	1	IRQ1	0x44
16	0	IRQ0	0x40
15	-1	SysTick	0x3C
14	-2	PendSV	0x38
13		Reserved	
12			
11	-5	SVCall	0x2C
10			
9			
8			
7		Reserved	
6			
5			
4			
3	-13	HardFault	0x10
2	-14	NMI	0x0C
1		Reset	0x08
		Initial SP value	0x04
			0x00

Контроллер прерываний. Основные LL методы

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

Репозиторий

https://github.com/edosedgar/stm32f0_ARM