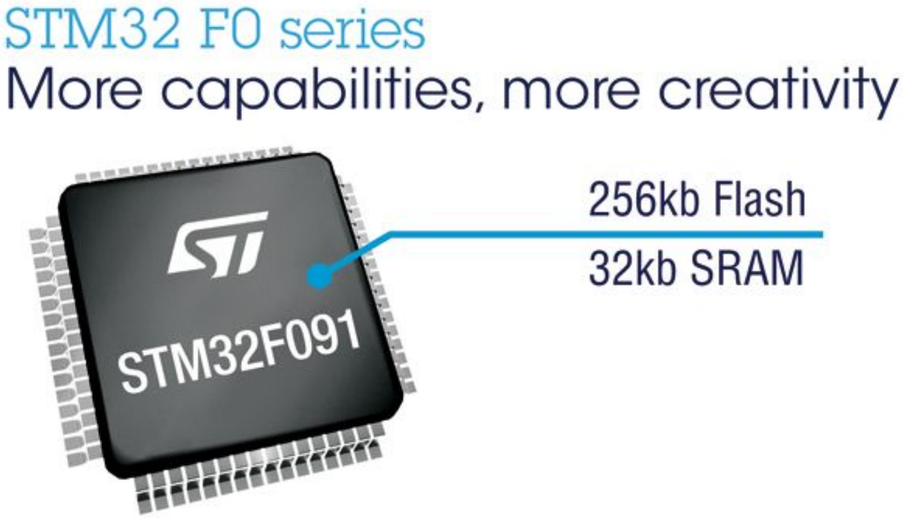
STM32 course

Getting into RCC internals and touching FLASH

A few words about previous lecture

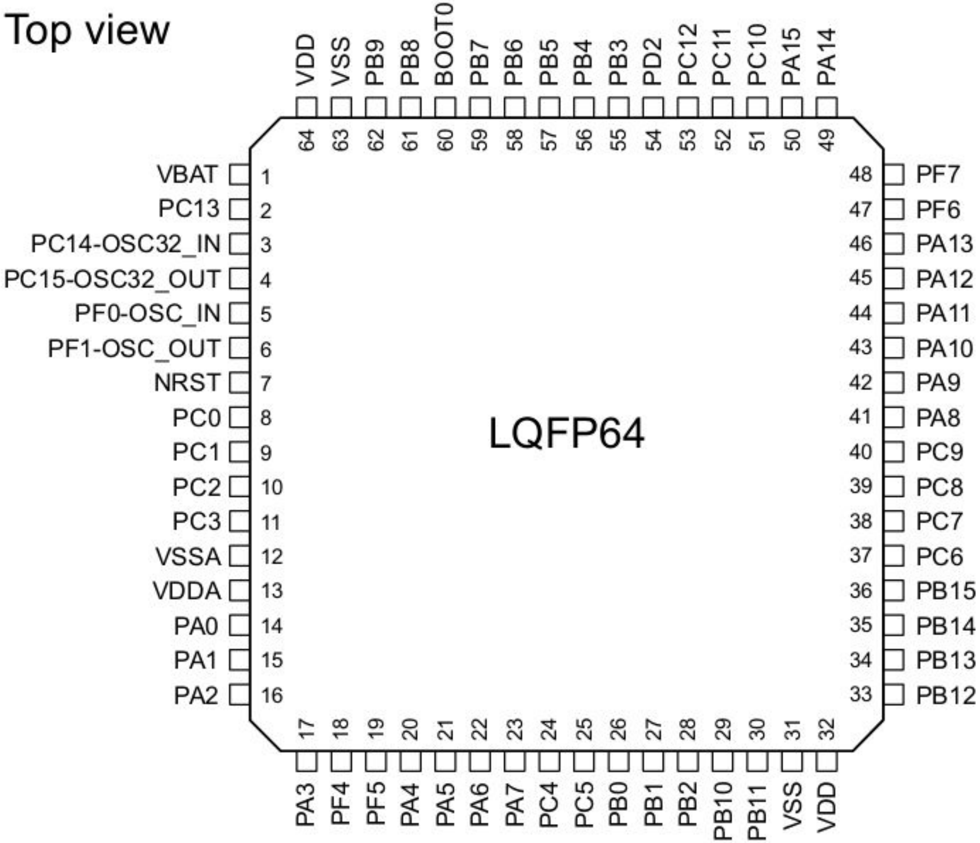
- Flip-flops, registers
- ALU
- SysTick
- STM32F0 SoC overview
- GPIO



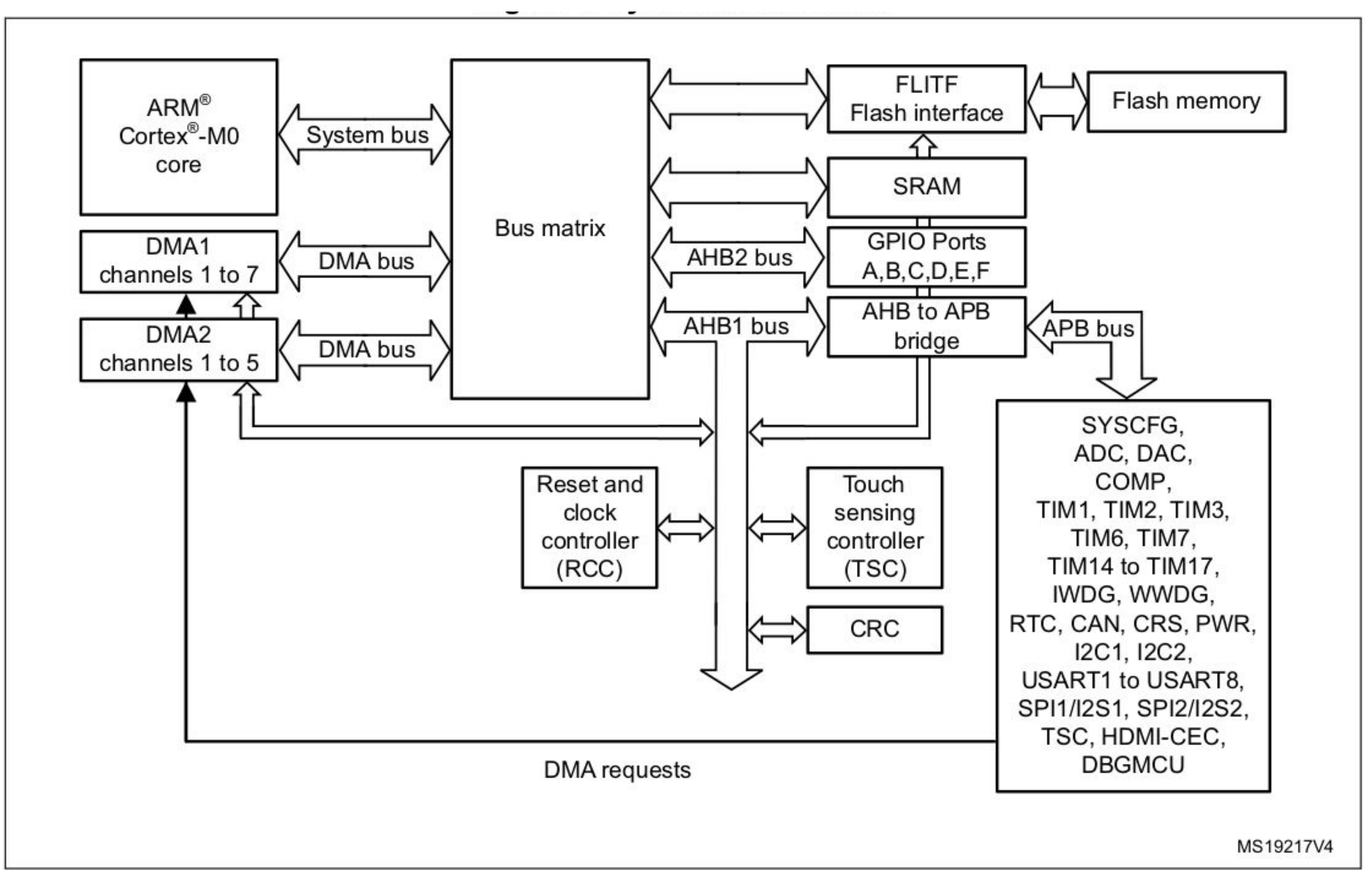
Outline

- Pinout
- STM32F0 SoC overview
- PWR
- RCC
- FLASH
- A small experiment with memory

PINOUT

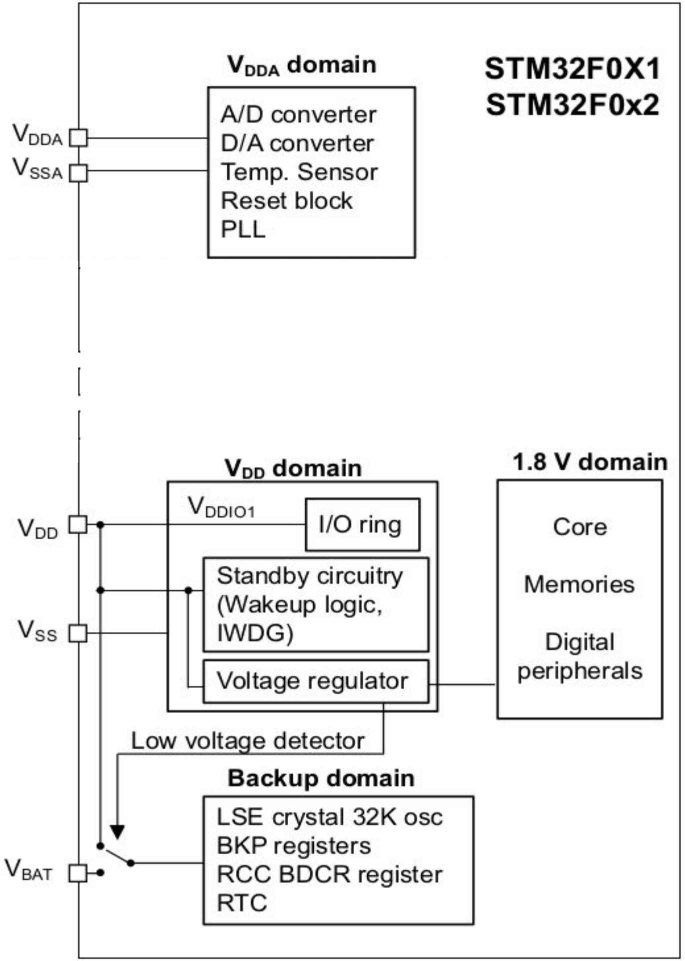

Top view
LQFP64
VBAT 1
PC13 2
PC14-OSC32_IN 3
PC15-OSC32_OUT 4
PF0-OSC_IN 5
PF1-OSC_OUT 6
NRST 7
PC0 8
PC1 9
PC2 10
PC3 11
VSSA 12
VDDA 13
PA0 14
PA1 15
PA2 16
17 PA3
18 PF4
19 PF5
20 PA4
21 PA5
22 PA6
23 PA7
24 PC4
25 PC5
26 PB0
27 PB1
28 PB2
29 PB10
30 PB11
31 VSS
32 VDD
33 PB12
34 PB13
35 PB14
36 PB15
37 PC6
38 PC7
39 PC8
40 PC9
41 PA8
42 PA9
43 PA10
44 PA11
45 PA12
46 PA13
47 PF6
48 PF7
49 PA14
50 PA15
51 PC10
52 PC11
53 PC12
54 PD2
55 PB3
56 PB4
57 PB5
58 PB6
59 PB7
60 BOOT0
61 PB8
62 PB9
63 VSS
64 VDD

STM32F0 SoC overview

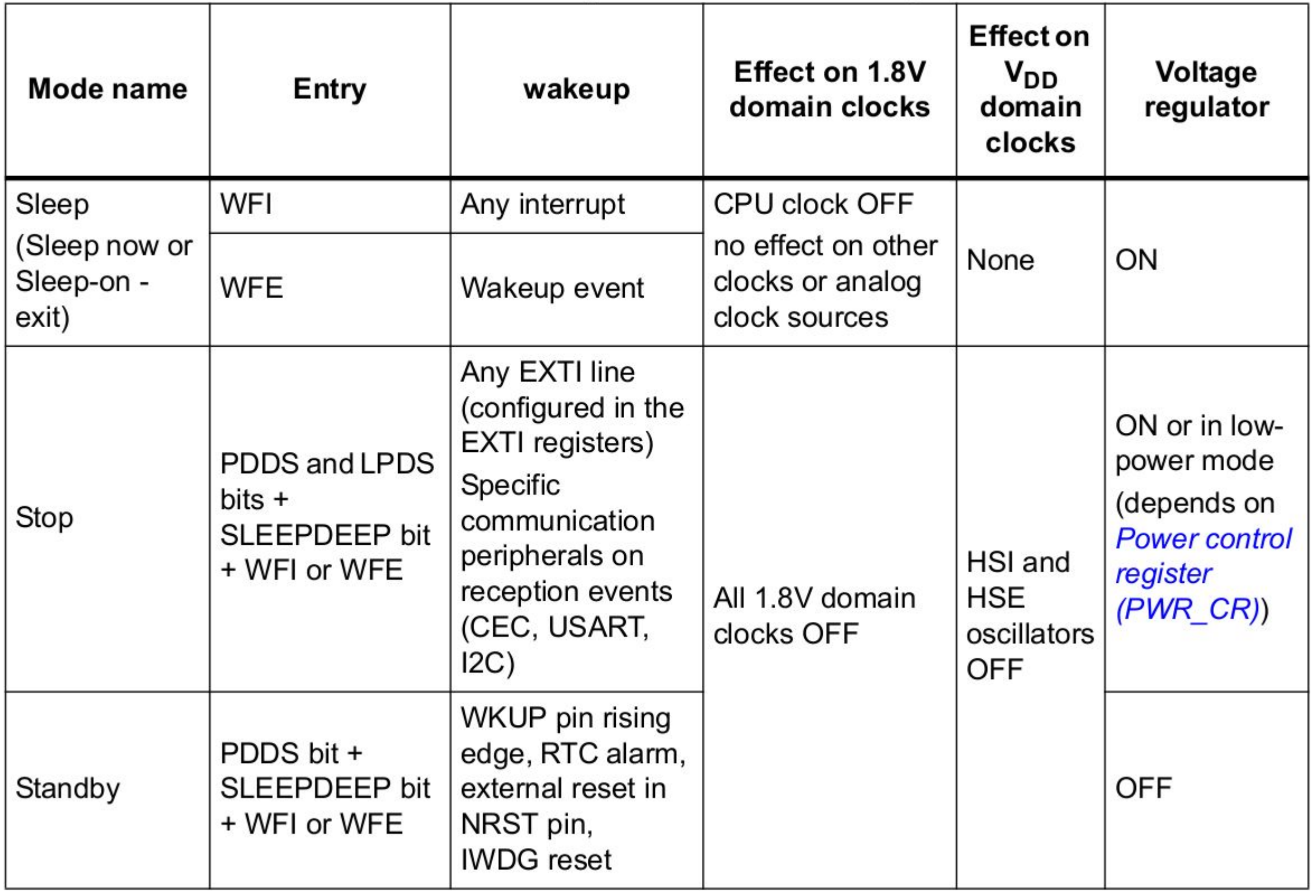

ARM® Cortex®-M0 core
System bus
Bus matrix
FLITF Flash interface
Flash memory
SRAM
DMA1 channels 1 to 7
DMA bus
AHB2 bus
GPIO Ports A,B,C,D,E,F
DMA2 channels 1 to 5
DMA bus
AHB1 bus
AHB to APB bridge
APB bus
Reset and clock controller (RCC)
Touch sensing controller (TSC)
CRC
SYSCFG, ADC, DAC, COMP, TIM1, TIM2, TIM3, TIM6, TIM7, TIM14 to TIM17, IWDG, WWDG, RTC, CAN, CRS, PWR, I2C1, I2C2, USART1 to USART8, SPI1/I2S1, SPI2/I2S2, TSC, HDMI-CEC, DBGMCU
DMA requests
MS19217V4

PWR (Power control)

- Independent A/D and D/A converter supply and reference voltage
- Battery backup domain
- Voltage regulator (core, memories and digital peripherals)
  - Run Mode (full power, 1.8V)
  - Stop Mode (only registers and RAM)
  - Standby Mode (only RTC)



PWR. Low-power modes

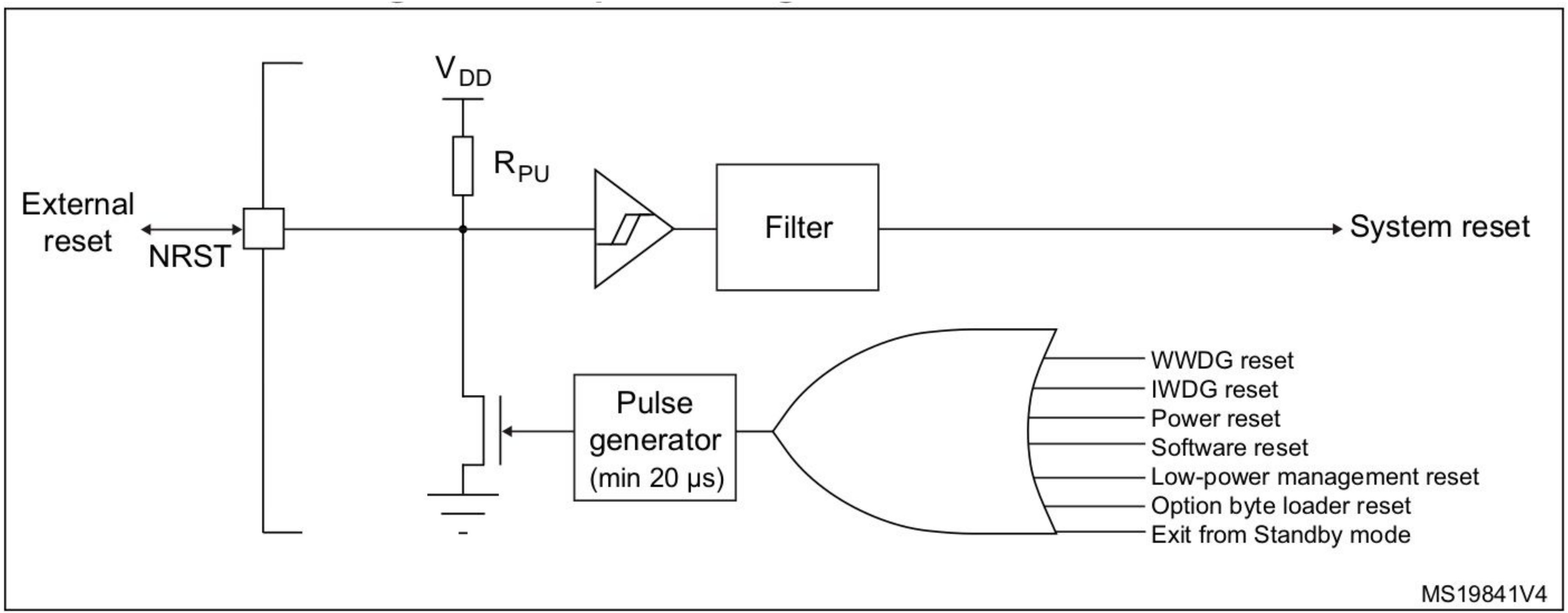


Mode name	Entry	wakeup	Effect on 1.8V domain clocks	Effect on V_{DD} domain clocks	Voltage regulator
Sleep (Sleep now or Sleep-on - exit)	WFI	Any interrupt	CPU clock OFF no effect on other clocks or analog clock sources	None	ON
	WFE	Wakeup event			
Stop	PDDS and LPDS bits + SLEEPDEEP bit + WFI or WFE	Any EXTI line (configured in the EXTI registers) Specific communication peripherals on reception events (CEC, USART, I2C)	All 1.8V domain clocks OFF	HSI and HSE oscillators OFF	ON or in low-power mode (depends on *Power control register (PWR_CR)*)
Standby	PDDS bit + SLEEPDEEP bit + WFI or WFE	WKUP pin rising edge, RTC alarm, external reset in NRST pin, IWDG reset			OFF

PWR. Power control registers

- Power control register (PWR_CR)
    - DBP (RTC and BKP registers write-protection)
    - PLS[2:0] (Power voltage detector threshold)
    - PVDE (turn on/off voltage detector)
    - CSBF, CWUF, PDDS, LPDS
- Power control/status register (PWR_CSR)
    - EWUPx and other status bits

RCC (Reset and clock control). Reset



RCC (Reset and clock control). Reset

- Software reset (NVIC_SystemReset())
- Low-power management reset
    - while entering Standby mode
    - while entering Stop mode
- Option byte loader reset (to launch the option byte loading by software)
- RTC domain reset (affects LSE, the RTC, the Backup registers and RCC)
    - Software reset (LL_RCC_ForceBackupDomainReset())
    - VDD power-up if VBAT has been disconnected when it was low

RCC (Reset and clock control). Clock Control

Various main clock sources:

- HSI 8 MHz RC oscillator clock
- HSE oscillator clock
- PLL clock

Additional clock sources:

- 40 kHz low speed internal RC (LSI RC)
- 32.768 kHz low speed external crystal (LSE crystal)
- 14 MHz high speed internal RC (HSI14) dedicated for ADC

RCC (Reset and clock control). Clock Control

Dangerous! Big pic! Switch to Reference Manual, p. 97

RCC (Reset and clock control). Clock Control

Clock Security System (CSS)

- Automatic detection failure with
  - NMI generation
  - Break input to Timers -> critical apps such as motor control put in a safe state
- Backup clock is HSI -> app doesn’t stop in case of crystal failure

RCC (Reset and clock control). Clock Control

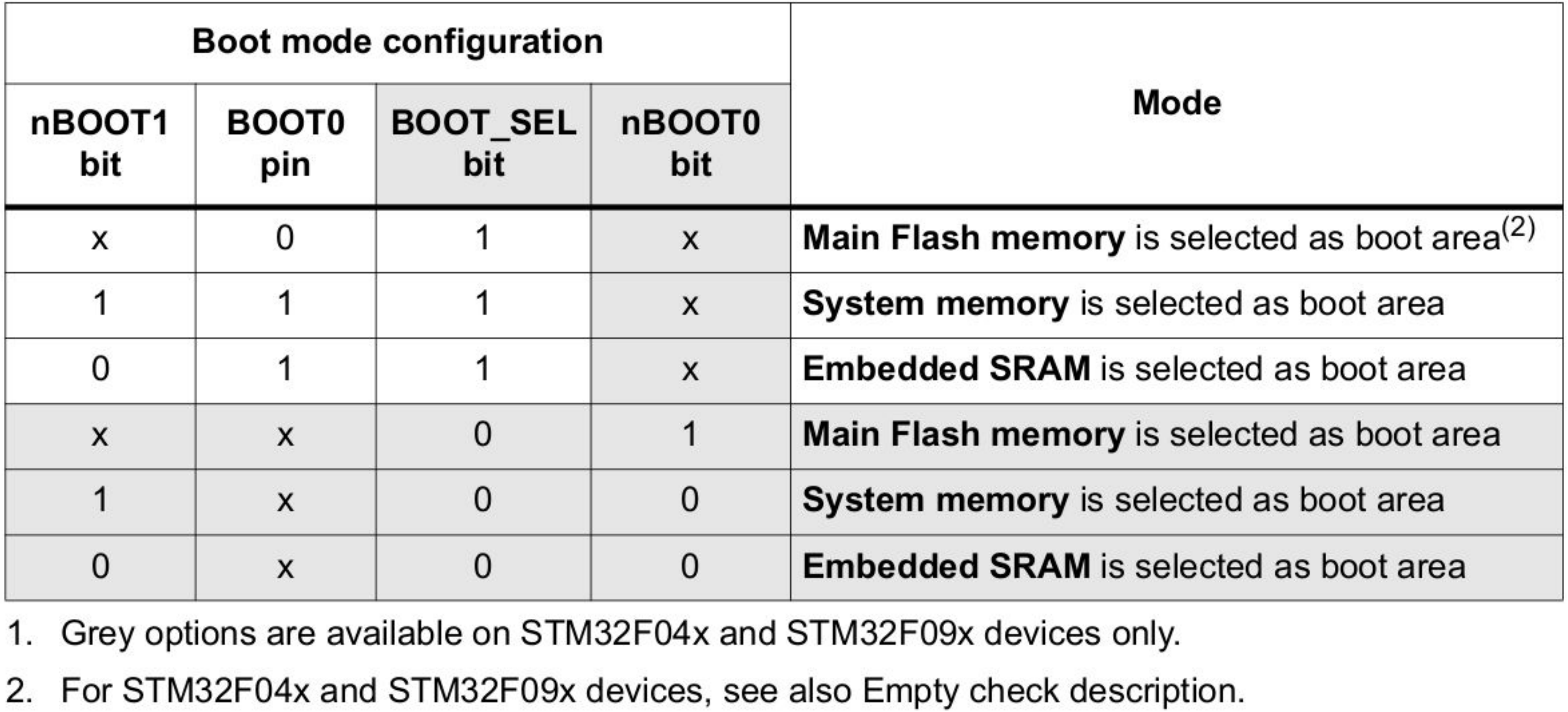
- Clock control register (RCC_CR)
  - PLLRDY, PLLON, CSSON, HSEBYP, HSERDY, HSERDY, HSIRDY, HSION, etc
- Clock configuration register (RCC_CFGR)
  - MCO[3:0] - Microcontroller clock output
  - PLLMUL[3:0] - PLL multiplication factor
  - PLLSRC[1:0] - PLL input clock source
  - PPRE[2:0] - PCLK prescaler (APB output freq)
  - HPRE[3:0] - HCLK prescaler (AHB output freq)
  - SWS[1:0], SW[1:0] - system clock
- Clock interrupt register (RCC_CIR)
- APB peripheral reset register [1,2] (RCC_APB[1,2]RSTR)
- AHB peripheral clock enable register (RCC_AHBENR)
- APB peripheral clock enable register [1,2] (RCC_APB[1,2]ENR)
- Clock configuration register 2 (RCC_CFGR2)
  - PREDIV[3:0] PREDIV division factor
- Clock control register 2 (RCC_CR2) (HSI14 configuration)

FLASH

- 64 Kbyte of Flash Memory
- Memory organization
  - Main Flash memory block
  - Information block (or System)
  - Up to 2 x 8 byte for the option byte
- Read interface with prefetch buffer (3 items)
  - The Cortex ® -M0 fetches the instruction over the AHB bus
  - Impact on the performance only when the wait state number is 1
- Option byte Loader
- Flash Program / Erase operation
- Read / Write protection

Flash area	Flash memory addresses	Size (byte)	Name	Description
Main Flash memory	0x0800 0000 - 0x0800 03FF	1 Kbyte	Page 0	Sector 0
	0x0800 0400 - 0x0800 07FF	1 Kbyte	Page 1	
	0x0800 0800 - 0x0800 0BFF	1 Kbyte	Page 2	
	0x0800 0C00 - 0x0800 0FFF	1 Kbyte	Page 3	
	. . .	. . .	. . .	. . .
	0x0800 7000 - 0x0800 73FF	1 Kbyte	Page 28	Sector 7 (1)
	0x0800 7400 - 0x0800 77FF	1 Kbyte	Page 29	
	0x0800 7800 - 0x0800 7BFF	1 Kbyte	Page 30	
	0x0800 7C00 - 0x0800 7FFF	1 Kbyte	Page 31	
	. . .	. . .	. . .	. . .
	0x0800 F000 - 0x0800 F3FF	1 Kbyte	Page 60	Sector 15
	0x0800 F400 - 0x0800 F7FF	1 Kbyte	Page 61	
	0x0800 F800 - 0x0800 FBFF	1 Kbyte	Page 62	
	0x0800 FC00 - 0x0800 FFFF	1 Kbyte	Page 63	
Information block	0x1FFF EC00 - 0x1FFF F7FF	3 Kbyte(2)	-	System memory
	0x1FFF C400 -0x1FFF F7FF	13 Kbyte(3)	-	System memory
	0x1FFF F800 - 0x1FFF F80F	2 x 8 byte	-	Option byte

FLASH. Boot Options

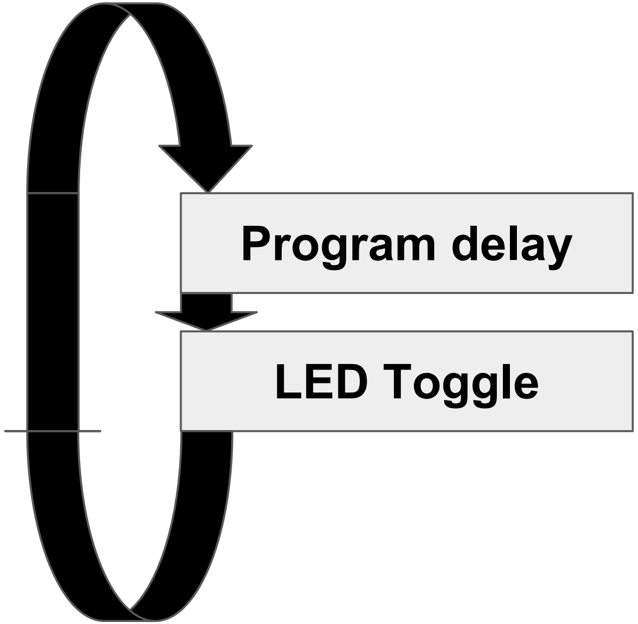


Boot mode configuration				Mode
nBOOT1 bit	BOOT0 pin	BOOT_SEL bit	nBOOT0 bit	
x	0	1	x	**Main Flash memory** is selected as boot area[2]
1	1	1	x	**System memory** is selected as boot area
0	1	1	x	**Embedded SRAM** is selected as boot area
x	x	0	1	**Main Flash memory** is selected as boot area
1	x	0	0	**System memory** is selected as boot area
0	x	0	0	**Embedded SRAM** is selected as boot area

1. Grey options are available on STM32F04x and STM32F09x devices only.

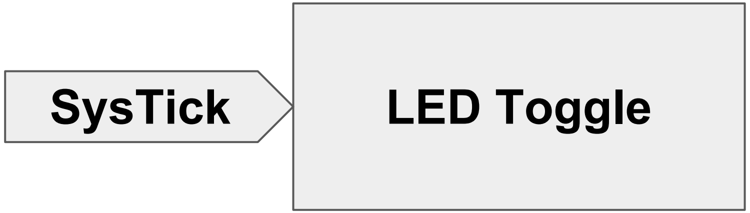
2. For STM32F04x and STM32F09x devices, see also Empty check description.

A small experiment with memory



Program delay

LED Toggle

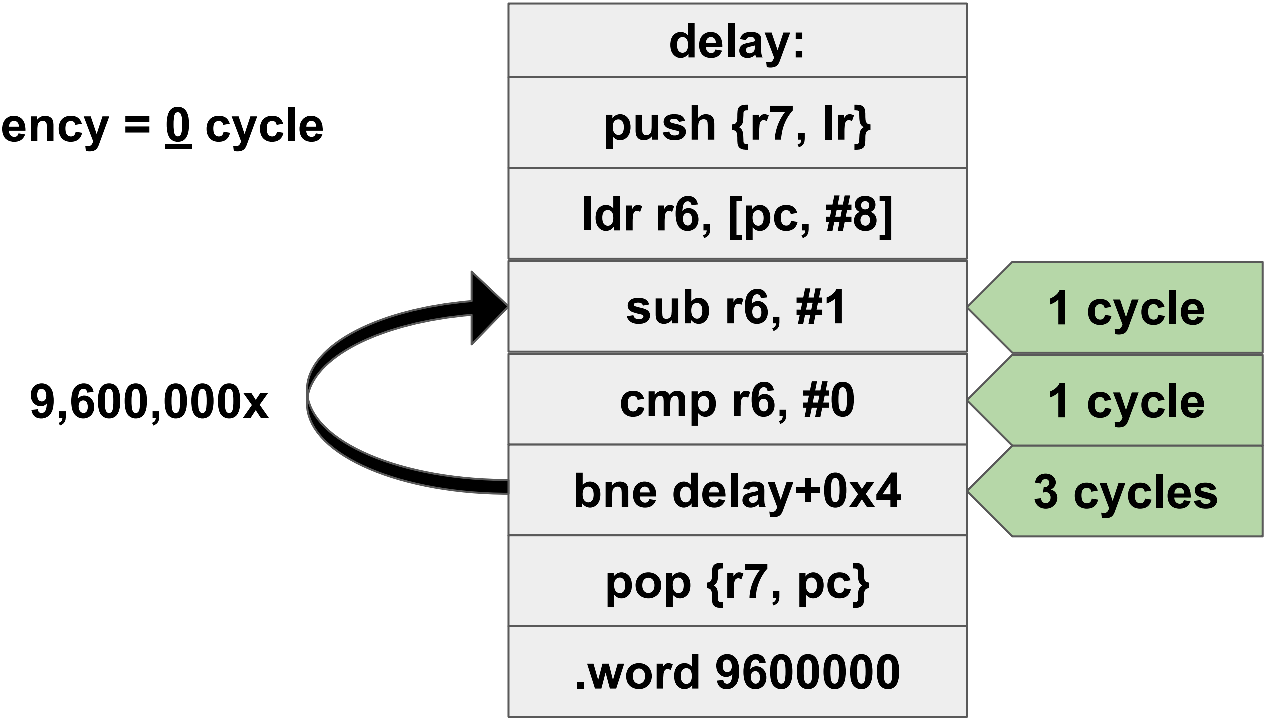


SysTick

LED Toggle

A small experiment with memory

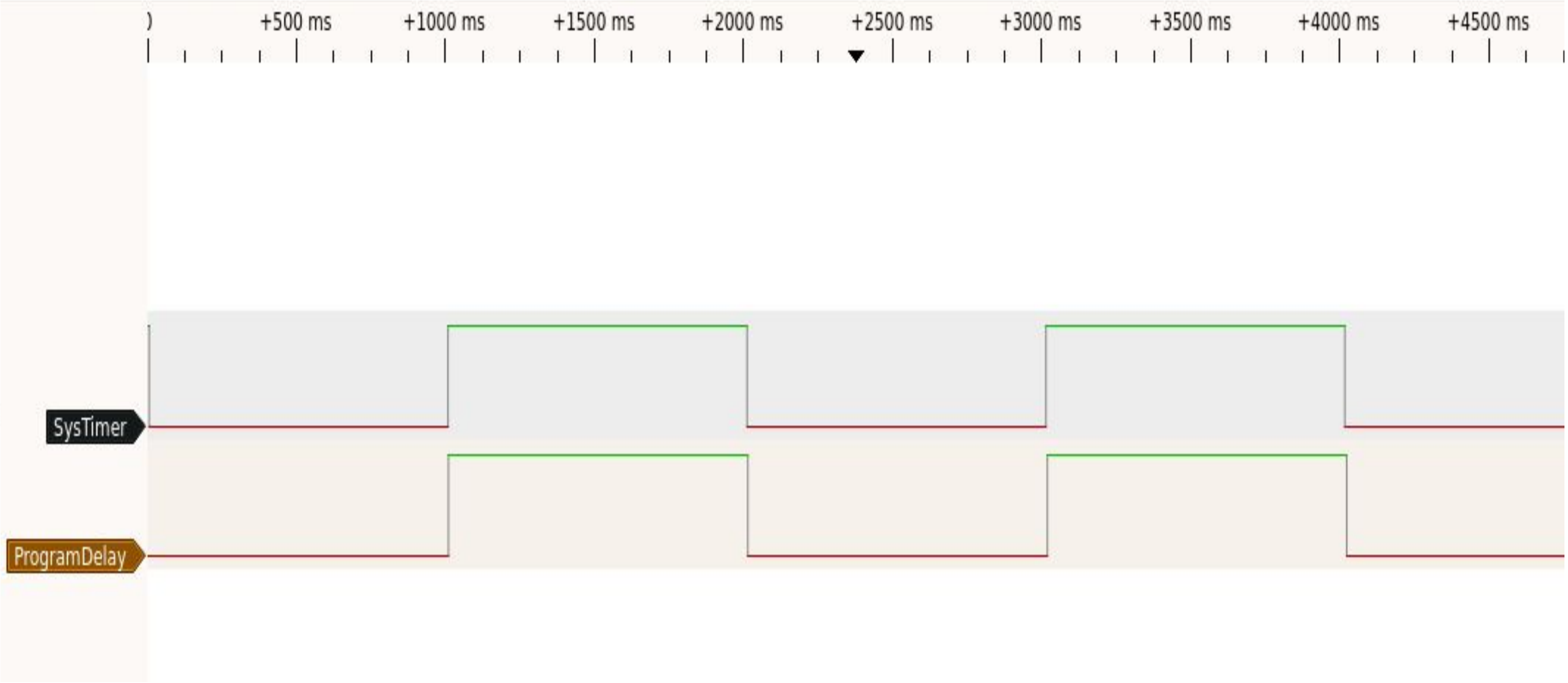
Memory Latency = <u>0</u> cycle



```
delay:
push {r7, lr}
ldr r6, [pc, #8]
sub r6, #1
cmp r6, #0
bne delay+0x4
pop {r7, pc}
.word 9600000
```

Delay = 9,600,000 x 5 cycles = 48,000,000 cycles = 1 sec

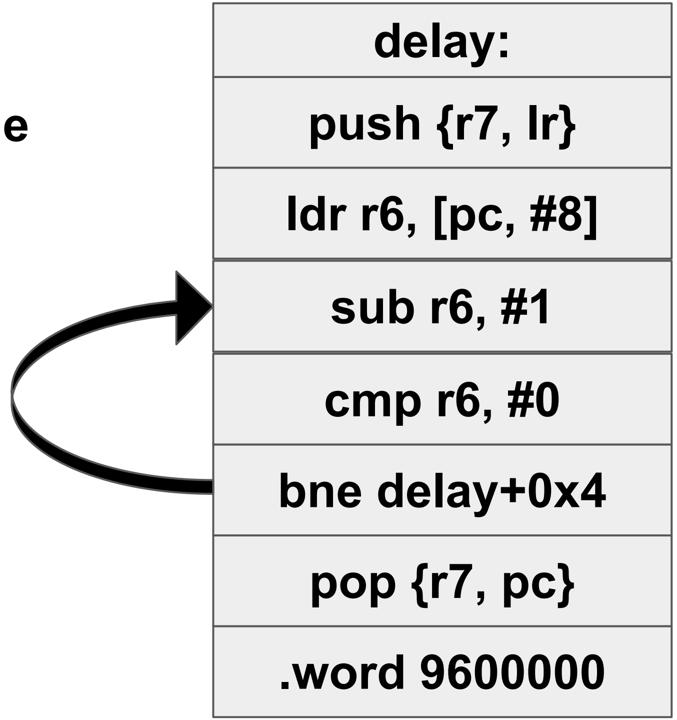
A small experiment with memory



A small experiment with memory

Memory Latency = 1 cycle

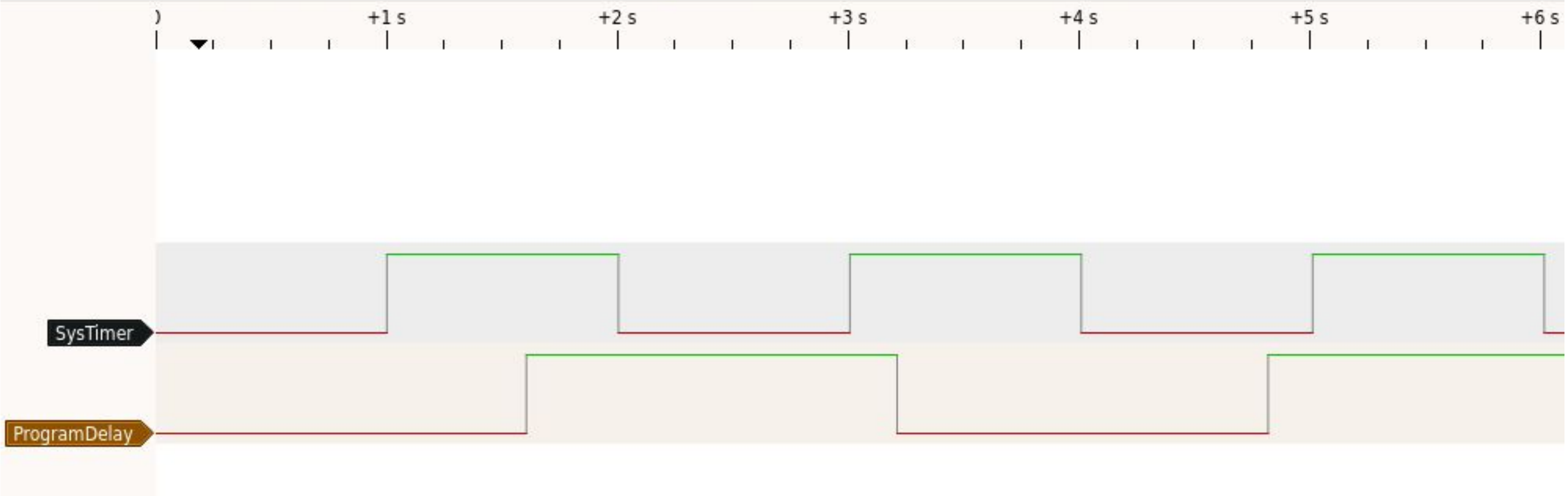
2 * 9,600,000x



```
delay:
push {r7, lr}
ldr r6, [pc, #8]
sub r6, #1
cmp r6, #0
bne delay+0x4
pop {r7, pc}
.word 9600000
```

Delay = 2 x 9,600,000 x 5 cycles = 96,000,000 cycles = 2 sec

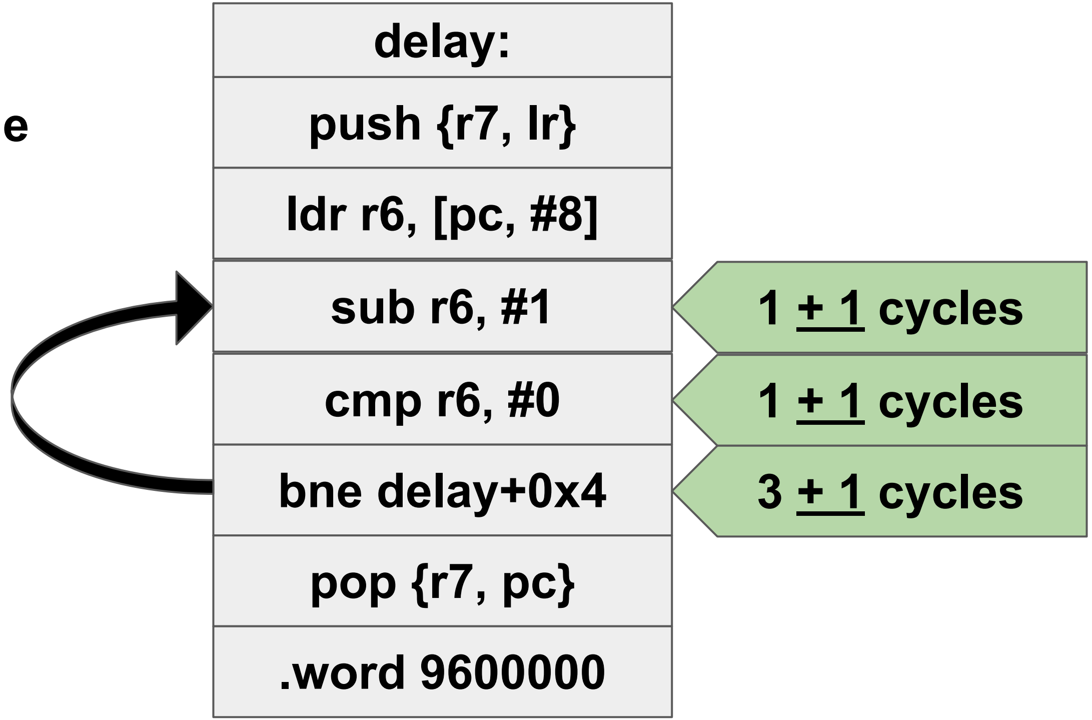
A small experiment with memory



Period = 3 sec (4 sec was expected). What’s wrong?

A small experiment with memory

Memory Latency = 1 cycle



```
delay:
push {r7, lr}
ldr r6, [pc, #8]
sub r6, #1
cmp r6, #0
bne delay+0x4
pop {r7, pc}
.word 9600000
```

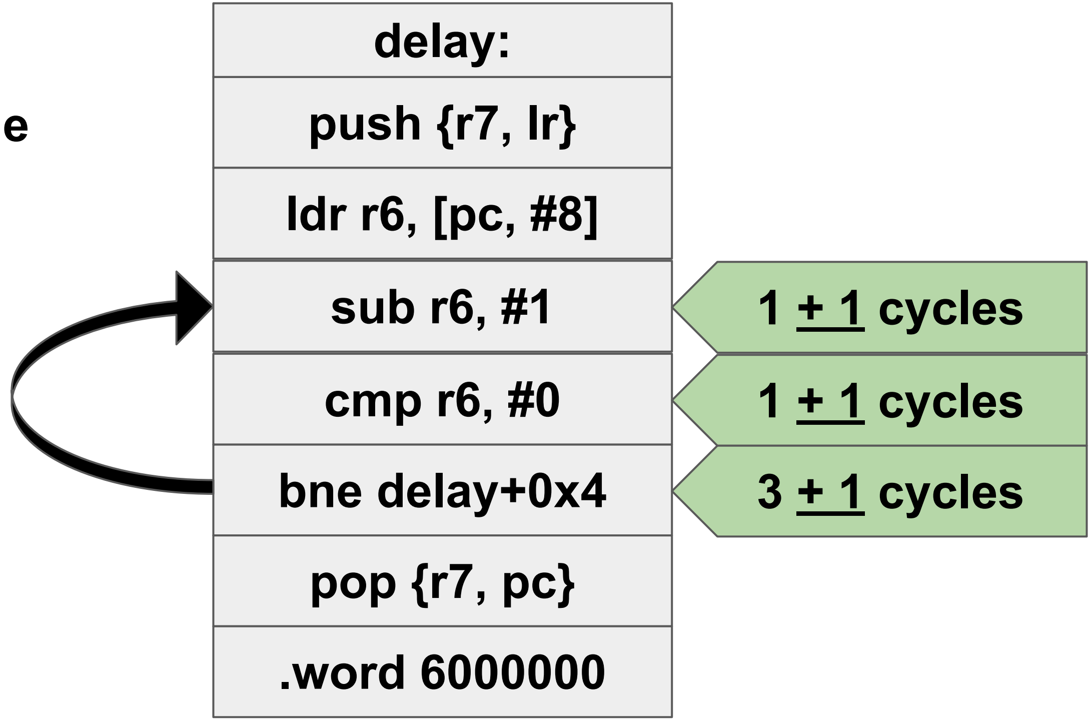
9,600,000x

Delay = 9,600,000 x 8 cycles = 76,800,000 cycles = 1,6 sec

A small experiment with memory

Memory Latency = 1 cycle

6,000,000x



```
delay:
push {r7, lr}
ldr r6, [pc, #8]
sub r6, #1
cmp r6, #0
bne delay+0x4
pop {r7, pc}
.word 6000000
```

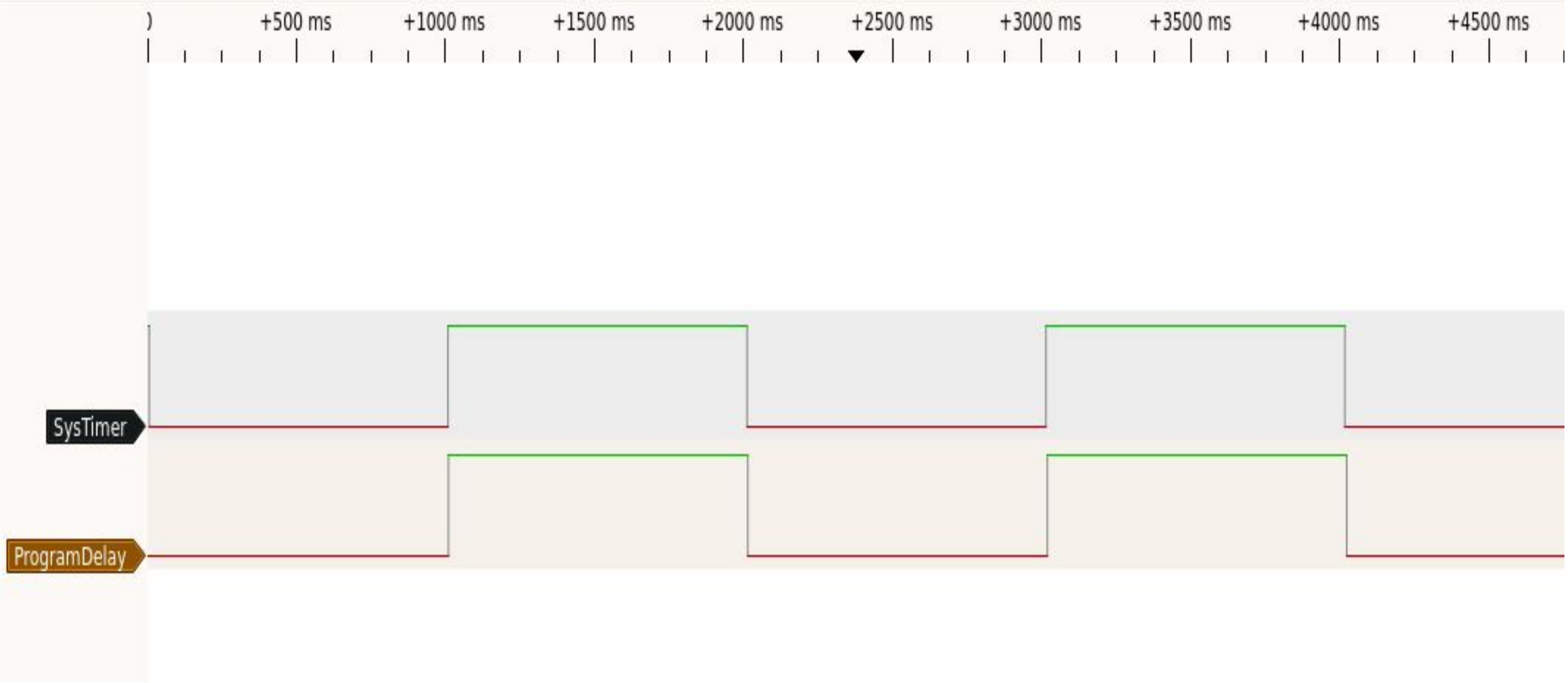
sub r6, #1: 1 + 1 cycles

cmp r6, #0: 1 + 1 cycles

bne delay+0x4: 3 + 1 cycles

Delay = 6,000,000 x 8 cycles = 48,000,000 cycles = 1 sec

A small experiment with memory


)
+500 ms
+1000 ms
+1500 ms
+2000 ms
+2500 ms
+3000 ms
+3500 ms
+4000 ms
+4500 ms
SysTimer
ProgramDelay