Clock Configuration

Lecture 08

Josh Brake
Harvey Mudd College

Outline

- Clock tree overview
- Phase locked loop (PLL) overview
- Activity: Configure the PLL to run the MCU at a desired frequency

Learning Objectives

By the end of this lecture you should be able to…

- Explain the different clock sources available on the MCU
- Know how to configure the phase locked loop on the MCU to generate a high-frequency clock source at a desired frequency.

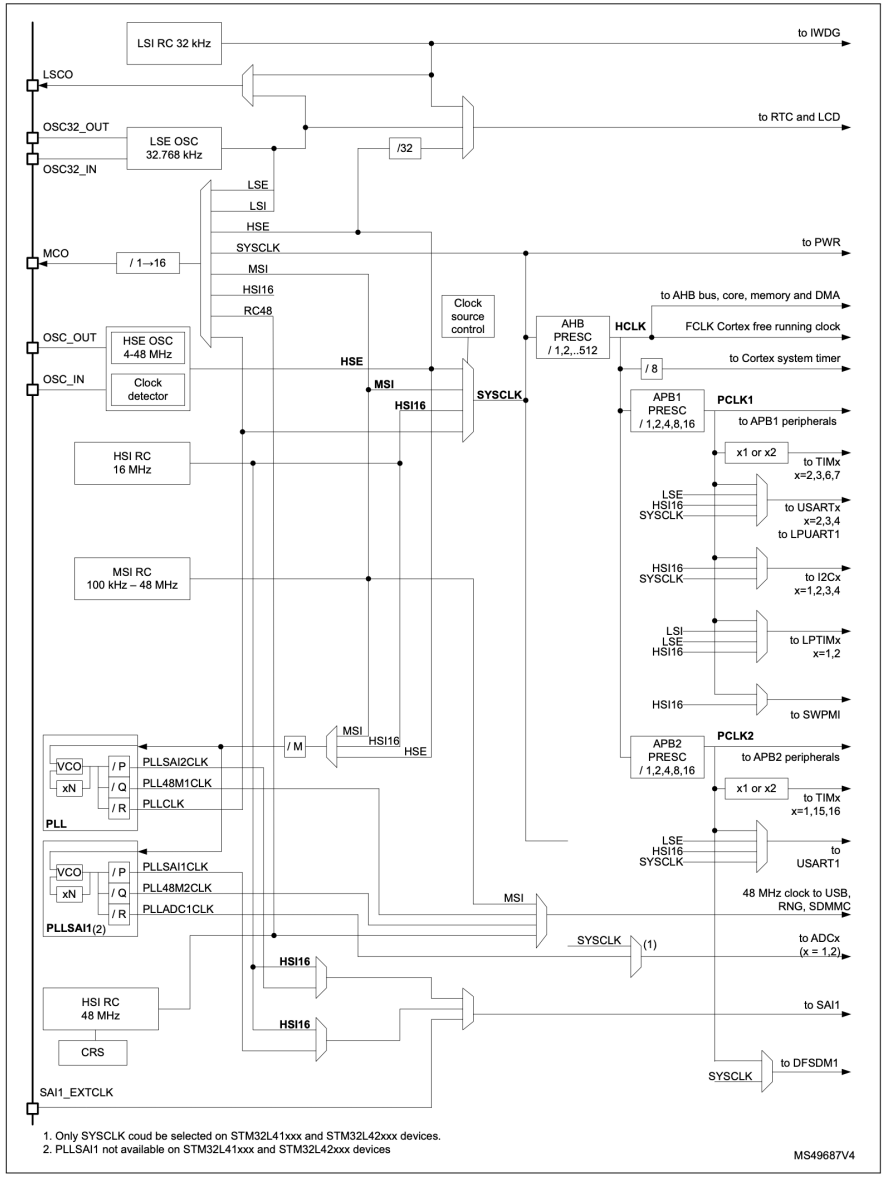
What is a clock tree?

The clock tree shows all the possible clock sources that can be used to drive the system busses and other peripherals. Four different sources can be used to drive the system clock (SYSCLK).

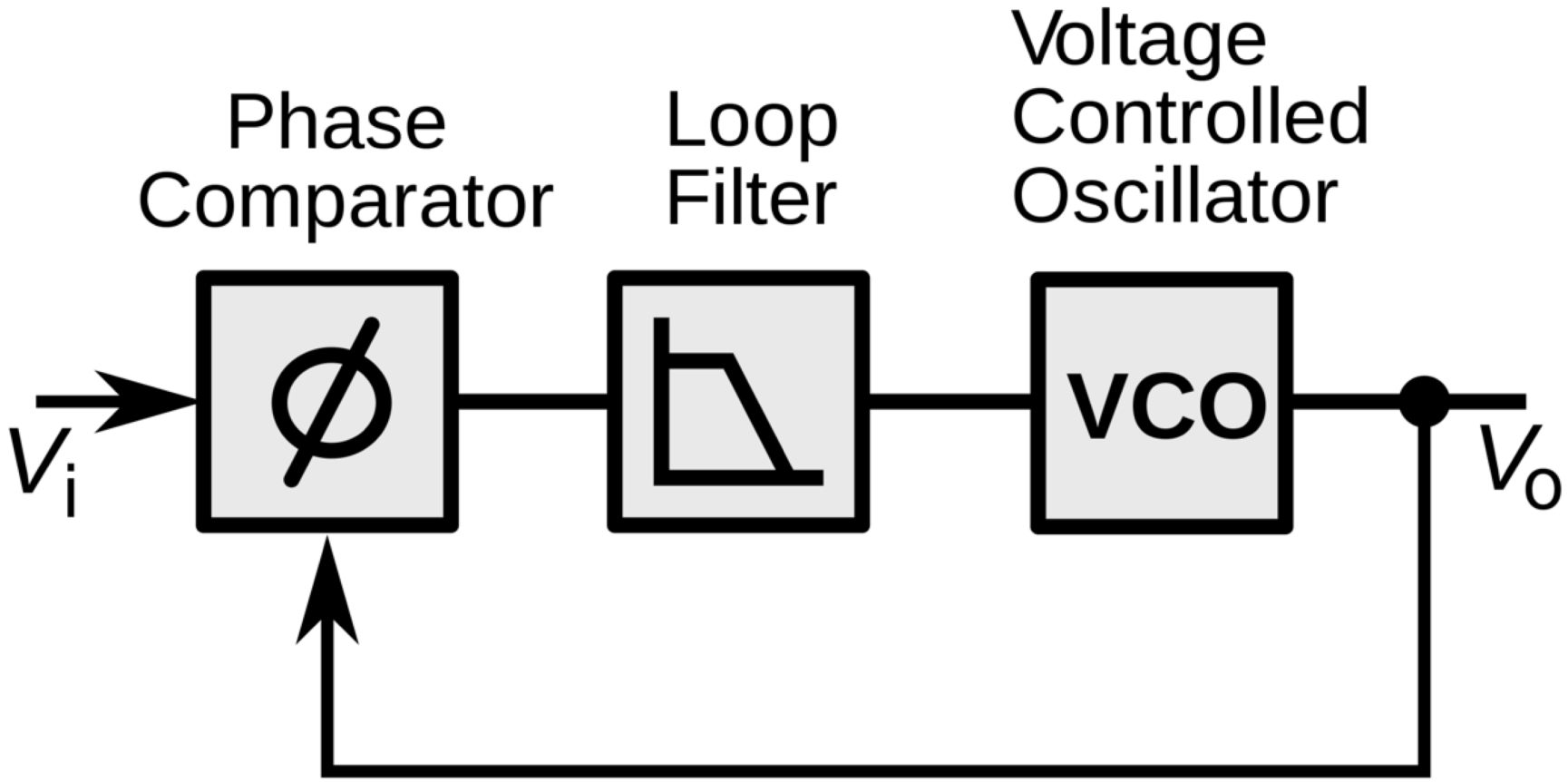
- HSI16 (high speed internal) 16 MHz
- MSI (multispeed internal) RC oscillator
- HSE (high speed external) 4 – 48 MHz
- PLL clock (phase locked loop)

MSI is set as default at reset configured at 4 MHz.

Figure 13. Clock tree



Phase Locked Loop (PLL)



Good info on Wikipedia page.

Steps to configure the PLL

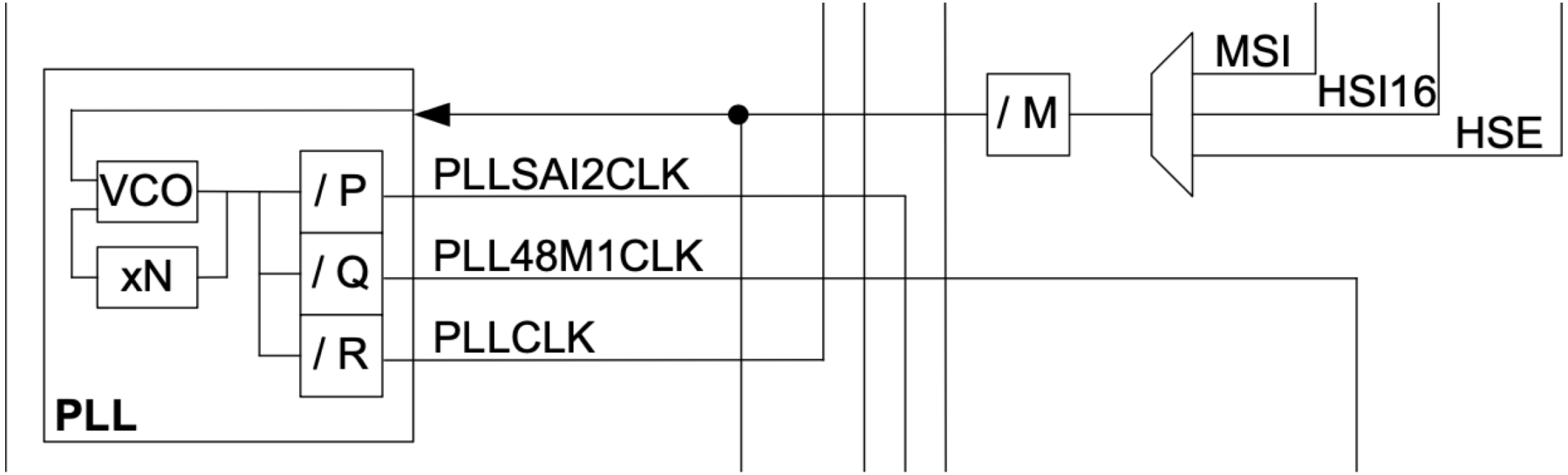
1. Select the input clock source
2. Set the main divisor M
3. Set the multiplier N
4. Set the divisor R
5. Turn on the PLL

Rules

PLL input frequency must be between 4 and 16 MHz. PLL output frequency must not exceed 80 MHz. VCO frequency must be between 64 and 344 MHz.

Phase Locked Loop

- Can be used to multiply clock speeds.
- Inputs must be between 4 and 16 MHz.



RM0394 p. 180

Activity: Configure PLL

RCC Device Driver

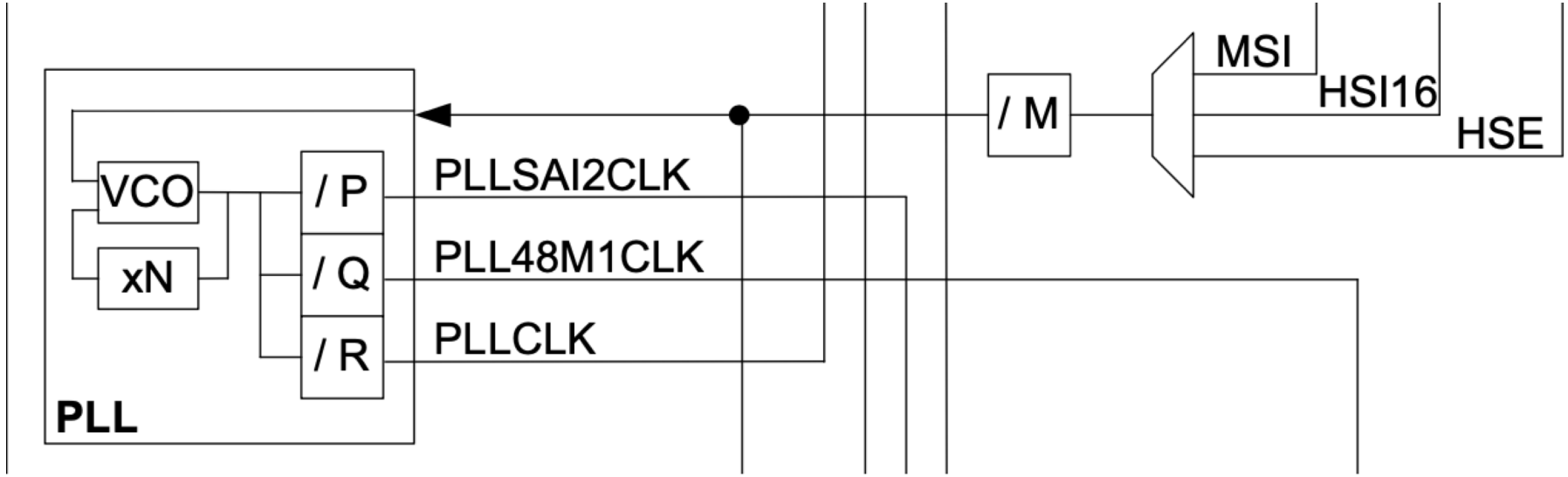
- Use the documentation to set up the PLL to generate an 80 MHz output clock signal and then switch it to be used as the system clock.
- Write a device driver `STM32L432KC_RCC.c/.h` that provides functions `configurePLL()` and `configureClock()` to accomplish this.

`configurePLL()` Worksheet: Values

The desired output frequency is 80 MHz. Using MSI as the input clock source (4 MHz) what are the required values and locations of the following parameters:

PLLCFGR

- PLL Clock Source = MSI
- M = 1
- N = 80
- R = 4



PLLCFGR

Bits 14:8 **PLLN[6:0]**: Main PLL multiplication factor for VCO

Set and cleared by software to control the multiplication factor of the VCO. These bits can be written only when the PLL is disabled.
VCO output frequency = VCO input frequency x PLLN with 8 =< PLLN =< 86
0000000: PLLN = 0 wrong configuration
0000001: PLLN = 1 wrong configuration
...
0000111: PLLN = 7 wrong configuration
0001000: PLLN = 8
0001001: PLLN = 9
...
1010101: PLLN = 85
1010110: PLLN = 86
1010111: PLLN = 87 wrong configuration
...
1111111: PLLN = 127 wrong configuration

Caution: The software has to set correctly these bits to assure that the VCO output frequency is between 64 and 344 MHz.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLPDIV[4:0]					PLLR[1:0]		PLL REN	Res.	PLLQ[1:0]		PLL QEN	Res.	Res.	PLLP	PLL PEN
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PLLN[7:0]							Res.	PLLM[2:0]			Res.	Res.	PLLSRC[1:0]	
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw			rw	rw

Bits 26:25 **PLLR[1:0]**: Main PLL division factor for PLLCLK (system clock)

Set and cleared by software to control the frequency of the main PLL output clock PLLCLK. This output can be selected as system clock. These bits can be written only if PLL is disabled.
PLLCLK output clock frequency = VCO frequency / PLLR with PLLR = 2, 4, 6, or 8
00: PLLR = 2
01: PLLR = 4
10: PLLR = 6
11: PLLR = 8

Caution: The software has to set these bits correctly not to exceed 80 MHz on this domain.

Bits 6:4 **PLLM:** Division factor for the main PLL and audio PLL (PLLSAI1) input clock

Set and cleared by software to divide the PLL PLLSAI1 input clock before the VCO. These bits can be written only when all PLLs are disabled.
VCO input frequency = PLL input clock frequency / PLLM with 1 <= PLLM <= 8
000: PLLM = 1
001: PLLM = 2
010: PLLM = 3
011: PLLM = 4
100: PLLM = 5
101: PLLM = 6
110: PLLM = 7
111: PLLM = 8

Caution: The software has to set these bits correctly to ensure that the VCO input frequency ranges from 4 to 16 MHz.

Bit 24 **PLLREN**: Main PLL PLLCLK output enable

Set and reset by software to enable the PLLCLK output of the main PLL (used as system clock).
This bit cannot be written when PLLCLK output of the PLL is used as System Clock.
In order to save power, when the PLLCLK output of the PLL is not used, the value of PLLREN should be 0.
0: PLLCLK output disable
1: PLLCLK output enable

`configurePLL()` Worksheet: Bits to Set

PLLCFGR

- PLL Clock Source = 0b01
- M = 0b000
- N = 80
- R = 0b01

PLL configuration register (RCC_PLLCFGR)

Address offset: 0x0C

Reset value: 0x0000 1000

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

This register is used to configure the PLL clock outputs according to the formulas:

- f(VCO clock) = f(PLL clock input) × (PLLN / PLLM)

- f(PLL_P) = f(VCO clock) / PLLP
- f(PLL_Q) = f(VCO clock) / PLLQ
- f(PLL_R) = f(VCO clock) / PLLR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLPDIV[4:0]					PLLR[1:0]		PLL REN	Res.	PLLQ[1:0]		PLL QEN	Res.	Res.	PLLP	PLL PEN
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PLLN[7:0]							Res.	PLLM[2:0]			Res.	Res.	PLLSRC[1:0]	
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw			rw	rw

Clock CR

6.4.1 Clock control register (RCC_CR)

Address offset: 0x00

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.	PLL SAI1 RDY(1)	PLL SAI1 ON(1)	PLL RDY	PLLON	Res.	Res.	Res.	Res.	CSS ON	HSE BYP	HSE RDY	HSE ON
				r	rw	r	rw					rs	rw	r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	HSI ASFS	HSI RDY	HSI KERON	HSION	MSIRANGE[3:0]				MSI RGSEL	MSI PLLEN	MSI RDY	MSION
				rw	r	rw	rw	rw	rw	rw	rw	rs	rw	r	rw

1. Not available on STM3L41xxx and STM32L42xxx devices

Bit 25 **PLLRDY**: Main PLL clock ready flag

Set by hardware to indicate that the main PLL is locked.

0: PLL unlocked

1: PLL locked

Bit 24 **PLLON**: Main PLL enable

Set and cleared by software to enable the main PLL.

Cleared by hardware when entering Stop, Standby or Shutdown mode. This bit cannot be reset if the PLL clock is used as the system clock.

0: PLL OFF

1: PLL ON

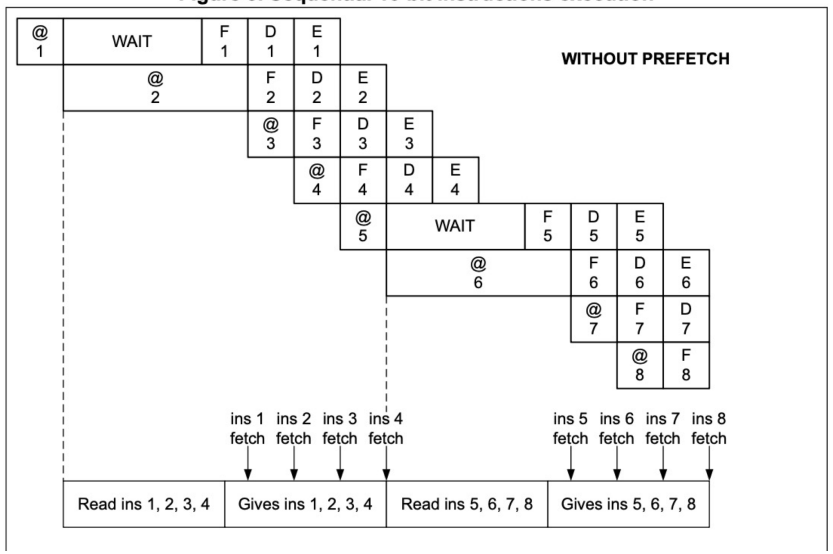
RCC Typedef Structure in `stm32l432xx.h`

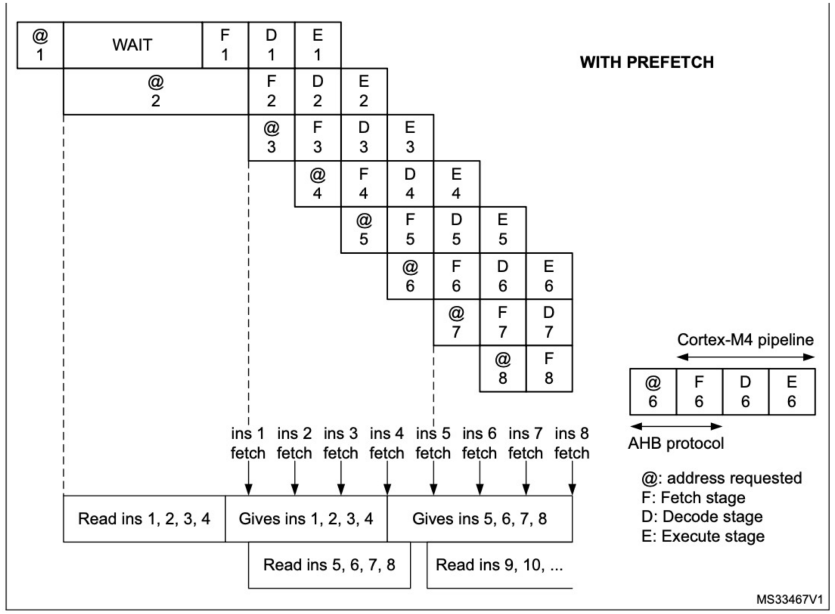
```
/**
  * @brief Reset and Clock Control
  */

typedef struct
{
  __IO uint32_t CR;          /*!< RCC clock control register,
  __IO uint32_t ICSCR;       /*!< RCC internal clock sources calibration register,
  __IO uint32_t CFGR;        /*!< RCC clock configuration register,
  __IO uint32_t PLLCFGR;     /*!< RCC system PLL configuration register,
  ...
  __IO uint32_t CRRCR;       /*!< RCC clock recovery RC register,
} RCC_TypeDef;
```

Flash Latency and the Adaptive real-time memory accelerator (ART Accelerator)

Figure 3. Sequential 16-bit instructions execution





When the code is not sequential (branch), the instruction may not be present in the currently used instruction line or in the prefetched instruction line. In this case (miss), the penalty in terms of number of cycles is at least equal to the number of wait states.

Wrap up

- Clocks are a critical part of the backbone of an MCU.
- The clock tree describes the different clocks available on the MCU and their configuration options
- Phase locked loops (PLLs) can be used to generate higher clock frequencies from a lower-frequency source.