COPYRIGHT

DISCLAIMER

This content is provided without warranty or representation of any kind. The use of the content is entirely at your own risk and Stellenbosch University (SU) will have no liability directly or indirectly as a result of this content.

The content must not be assumed to provide complete coverage of the particular study material. Content may be removed or changed without notice.

The video is of a recording with very limited post-recording editing. The video is intended for use only by SU students enrolled in the particular module.

UNIVERSITEIT
iYUNIVESITHI
STELLENBOSCH
UNIVERSITY

100
1918 · 2018

forward together · saam vorentoe · masiye phambili

Computer Systems / Rekenaarstelsels 245 - 2020

Lecture 12

Clock Signals
Klokseine

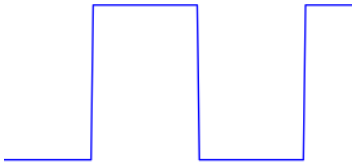
Dr Rensu Theart & Dr Lourens Visagie

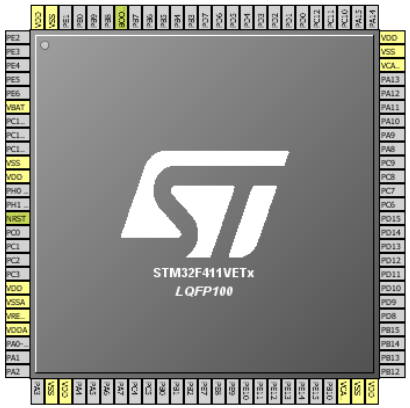
Lecture Overview

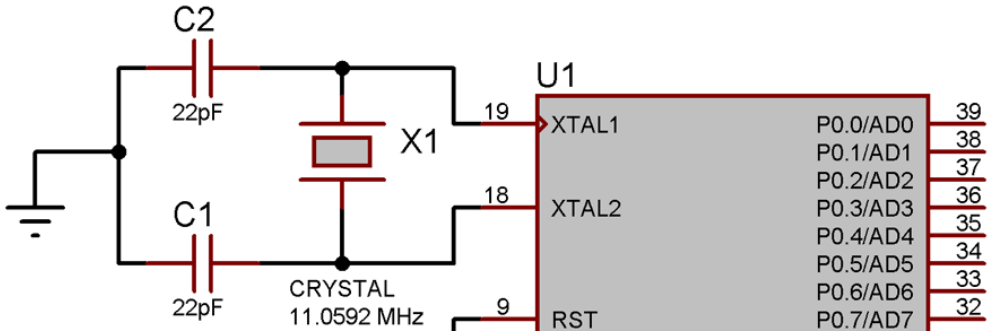
- Microcontroller clock input
- Types of clock sources
  - RC
  - Crystal
- Phase-locked-loops
- STM32F clocks
  - Configuration

Microcontroller clock input / Mikrobeheerder klok intree

- Every microcontroller needs a clock source – used by the CPU, memory, bus, and peripherals.
- The clock input is a periodic signal with (ideally) fixed frequency.
- It determines
  - How fast instructions execute
  - Rate of serial communication
  - Time between analog-to-digital samples

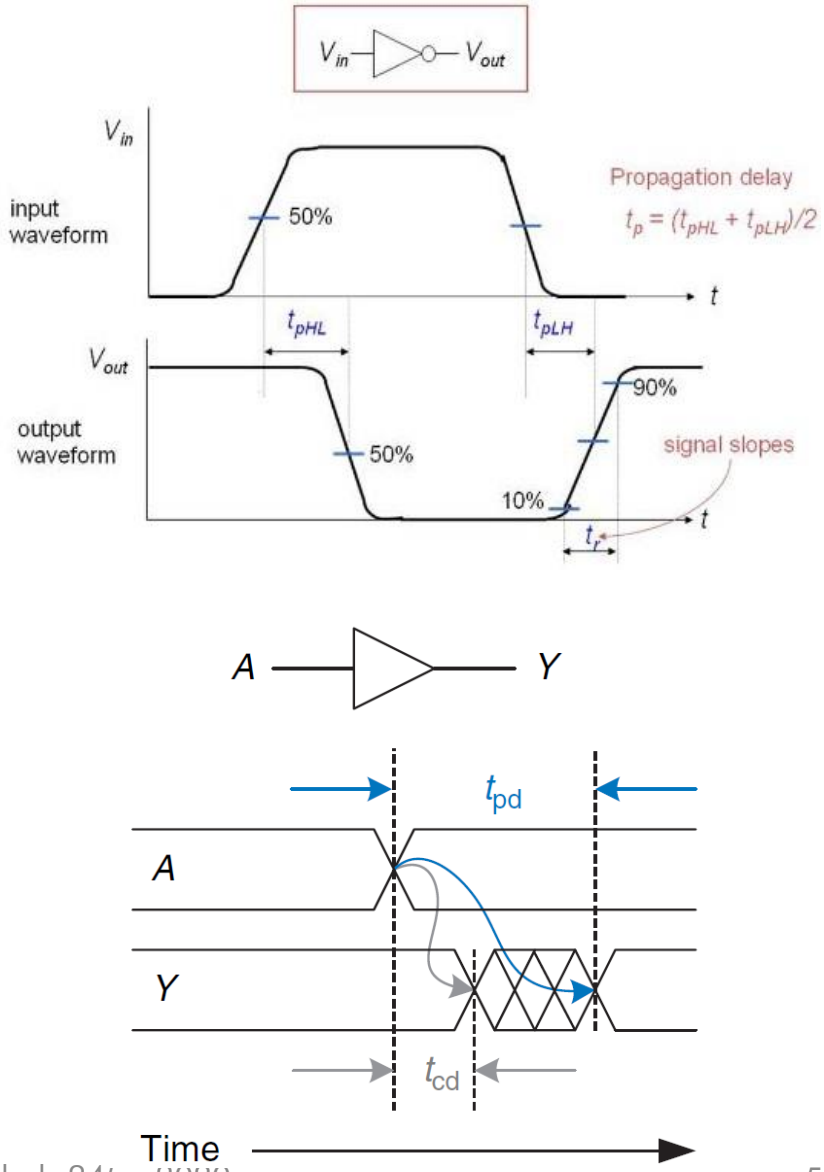






Why does a microcontroller need a clock input? / Waarom is 'n klok intree nodig?

- Microcontrollers execute **sequential** logic:
  - logic circuit's output depends not only on the present value of its input signals but on the sequence of past inputs - the input history.
  - Think flip-flops / finite state machines
- The logic gates (inside the microcontroller) require a finite amount of time to respond to changes to their inputs – called the **propagation delay**
  - Following a change in input voltage on a gate, the output signal level will be indeterminate for the duration of the propagation delay.
- Time between clock pulses must allow for all logic levels to settle to determinate levels – so that gate outputs have enough time so adjust to changes in inputs.



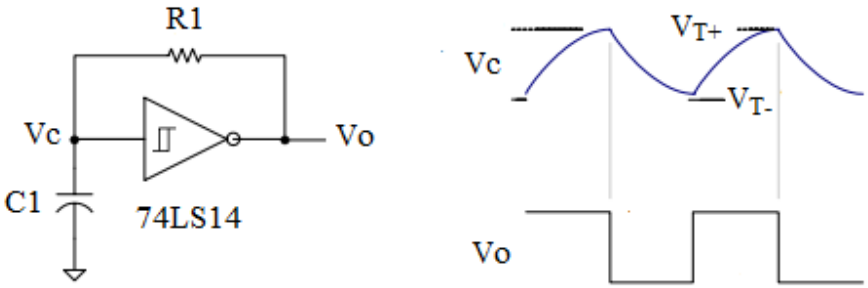
Types of clock sources / Verskillende bronne van klok seine

- Internal to microcontroller
  - Resistor–capacitor circuit (RC oscillator)
  - Phase-locked loop (PLL) for frequency multiplication
- External to microcontroller
  - CMOS clock
  - Crystal (Quartz)
  - Ceramic resonator
  - Resistor–capacitor (RC oscillator)

RC oscillator / Weerstand-kapasitor ossillator

- Cyclic charge and discharge of capacitor causes periodic variation in output voltage.
- Different ways to implement.

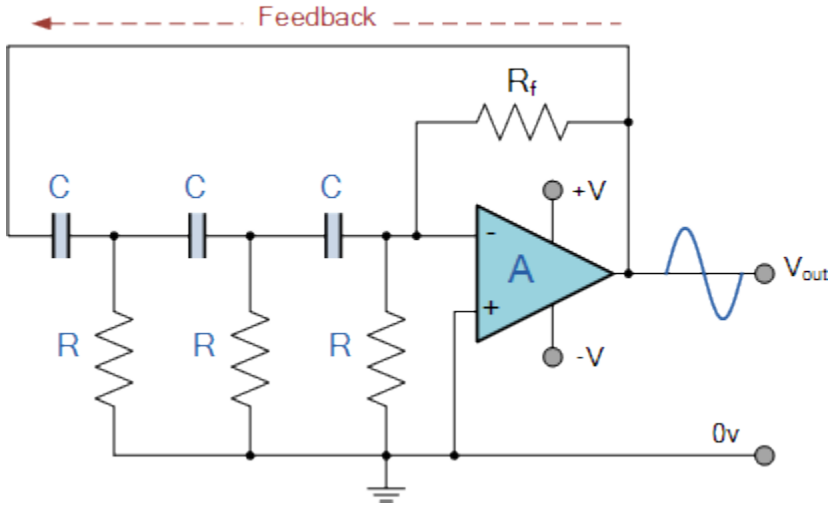
Schmitt trigger oscillator



$$f_o = \frac{0.8}{R_1 C_1}$$

(formula depends on the device used for the Schmitt trigger)

Phase-shift oscillator



$$f_r = \frac{1}{2\pi RC\sqrt{2N}}$$

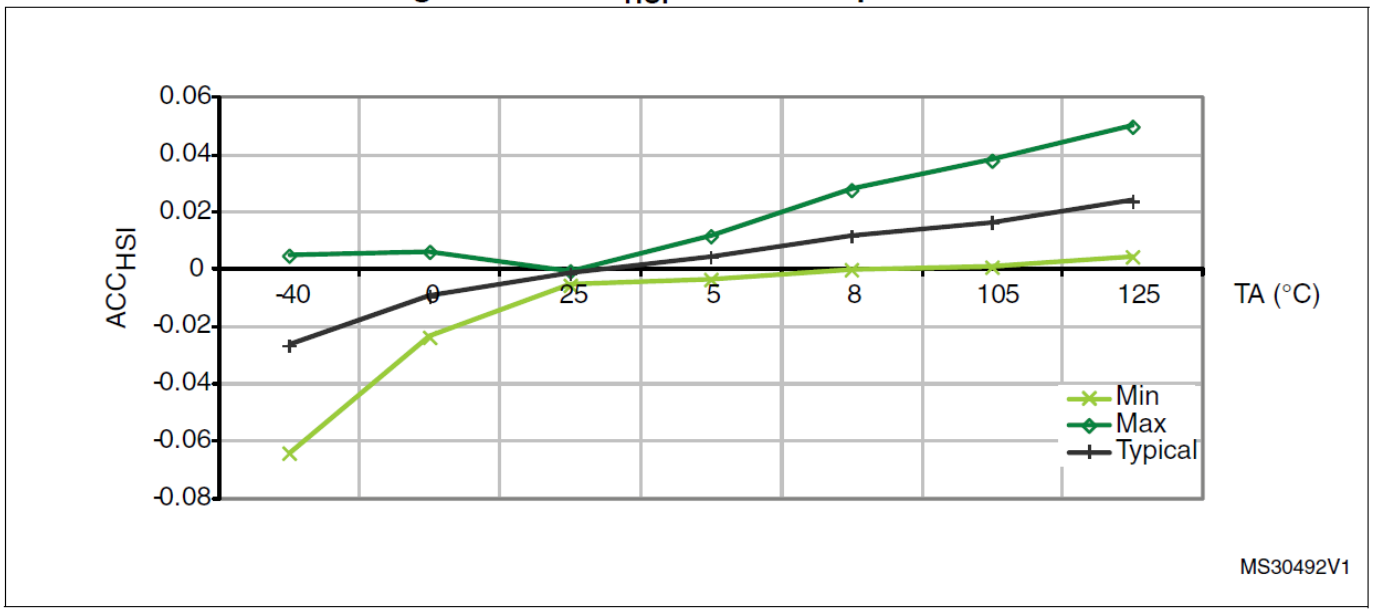
(N = number of RC-stages (3 in the above example))

RC oscillator / Weerstand-kapasitor ossillator

- Low-cost and simple
- Disadvantage: Lack of precision and stability.
  - Subject to manufacturing tolerances (1k resistor is not exactly 1k). And resistance and capacitance values determine the resonant frequency (lack of precision)
  - Resistance and capacitance tend to change with temperature – causes drift in frequency (instability)

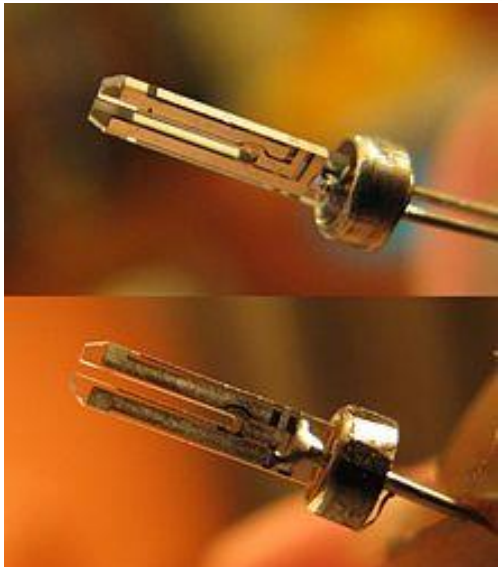
High-speed internal clock

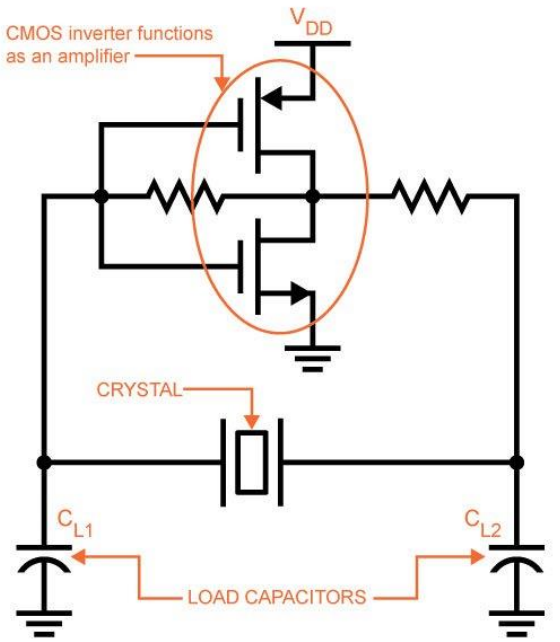
Figure 26. ACC_{HSI} versus temperature



Crystal oscillator / Kristal ossillator

- A crystal oscillator consists of a quartz crystal and the additional circuitry needed to generate a standard digital clock signal.
- Uses the mechanical resonance of a vibrating crystal of piezoelectric material (quartz) to create an electrical signal.
- The quartz crystal deforms (changes shape) as a result of electric field applied across the crystal (electrostriction).
- When the field is removed, the crystal goes back to original shape and creates an electric field as a result
  - Electric field can be used to generate a voltage
- Resonant frequency is a result of mechanical properties of the crystal (size, and how it was cut)
- Requires oscillator control circuit to amplify crystal output voltage and feed back to resonator.
- Displays less variation with temperature (but still varies).





Oscillator accuracy / Ossillator akkuraatheid

- The variation in frequency usually specified in 'ppm' units (parts-per-million).
- For instance, a quartz oscillator frequency is rated as 32.768 kHz +/- 20 ppm
- The frequency range is then

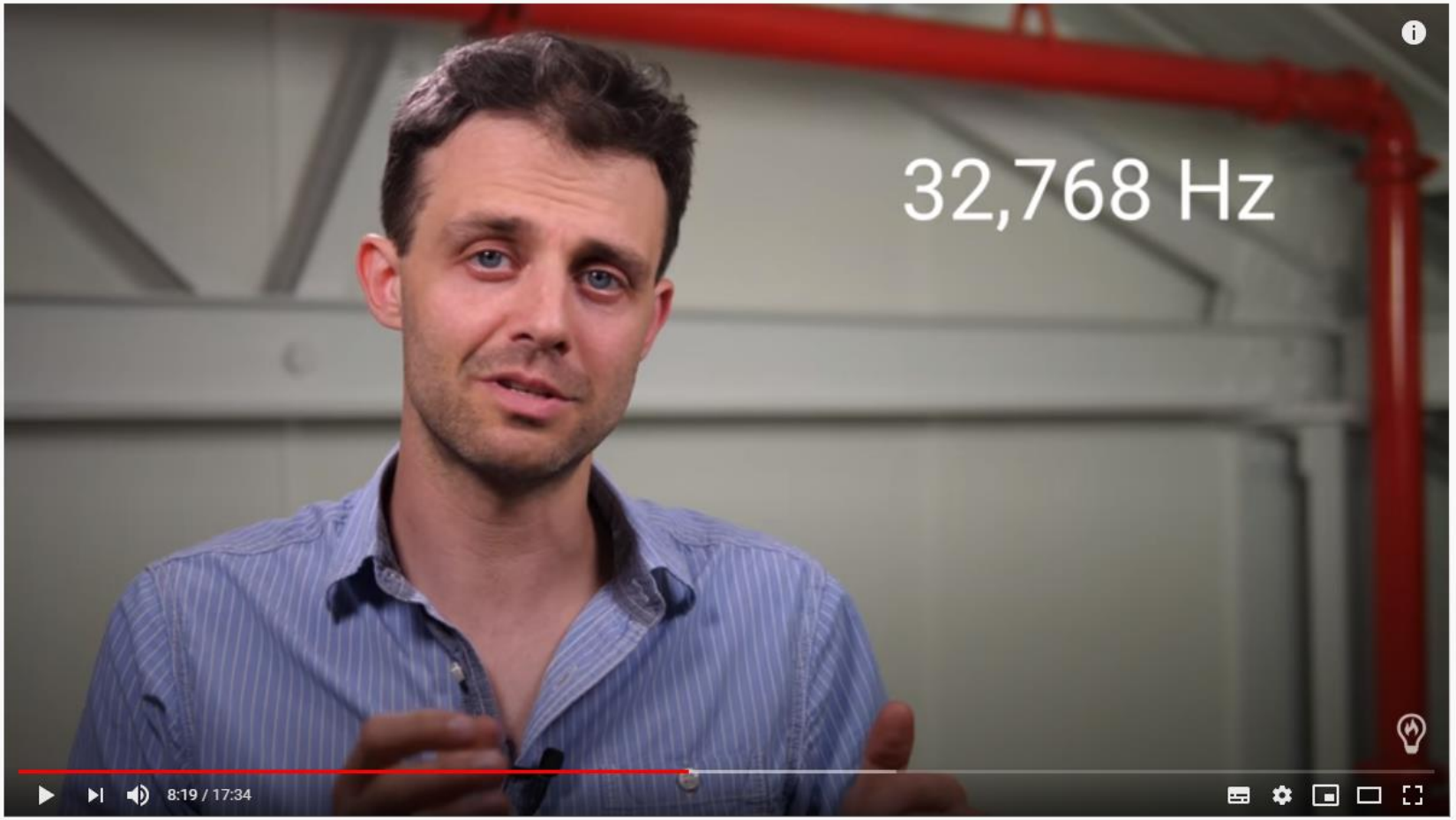
$$32.768\,kHz \pm \frac{32.768\,kHz * 20}{1{,}000{,}000}$$

- Or in other words the frequency may range from 32.7673 to 32.7687 kHz

- How accurate is it at keeping time?
  - The clock (built using this crystal oscillator) may lose up to 20s every 1,000,000s. It might be out by as much as 50s over a 1 month period.

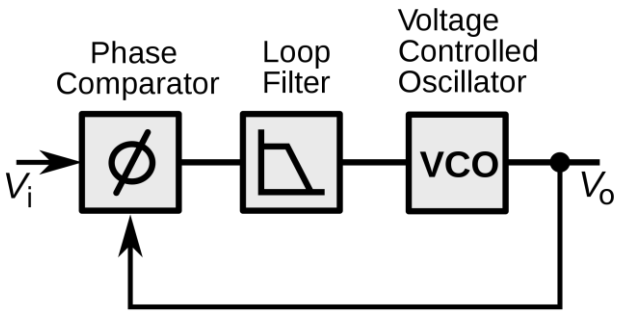
Crystal oscillator / Kristal ossillator

- https://www.youtube.com/watch?v=_2By2ane2I4

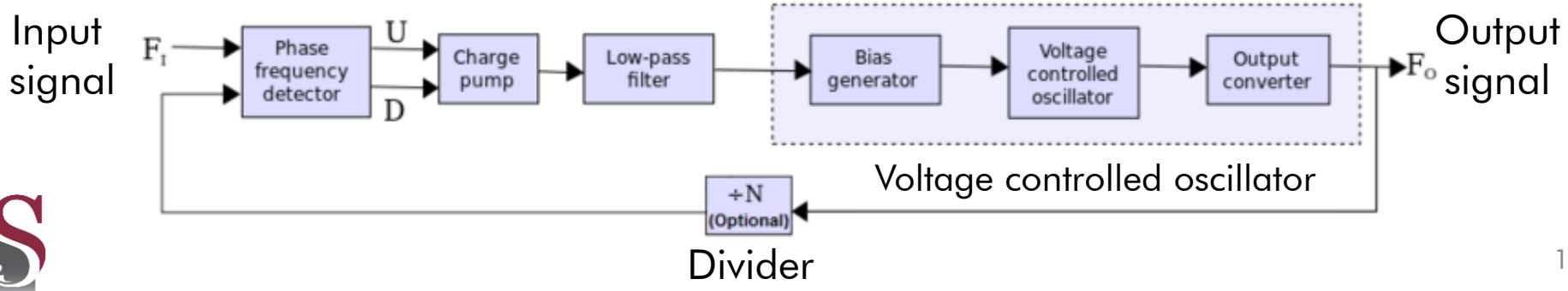


PLL and clock multiplication / Fase-sluit lus en klok vermenigvuldiging

- Crystals cannot resonate at GHz frequencies.
- Use **Phase-Locked-Loop (PLL)** with divider to create multiples of input frequency.
- Phase-Locked-Loop: Phase detector, filter and voltage-controlled-oscillator (VCO)

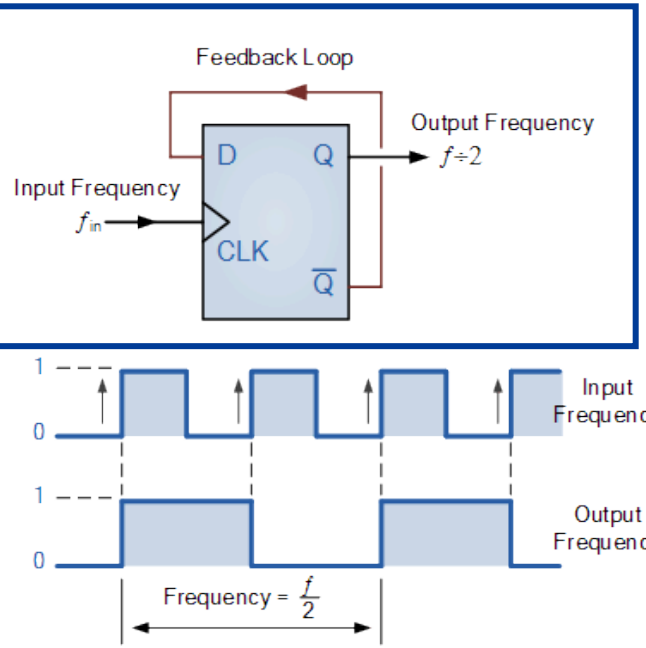


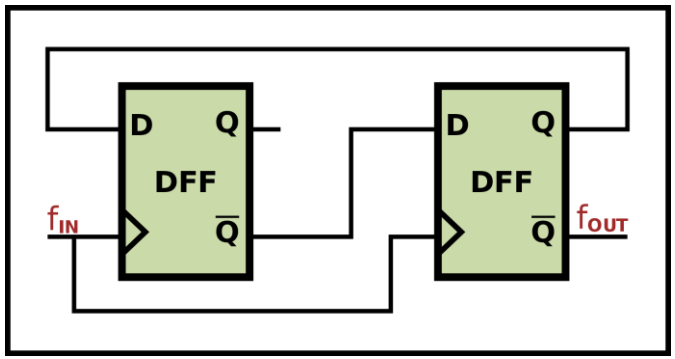
- The output is fed back to the input of the system, producing a negative feedback loop.
  - If the output phase drifts, the error signal will increase, driving the VCO phase in the opposite direction so as to reduce the error.
- Thus the output phase is locked to the phase at the other input.

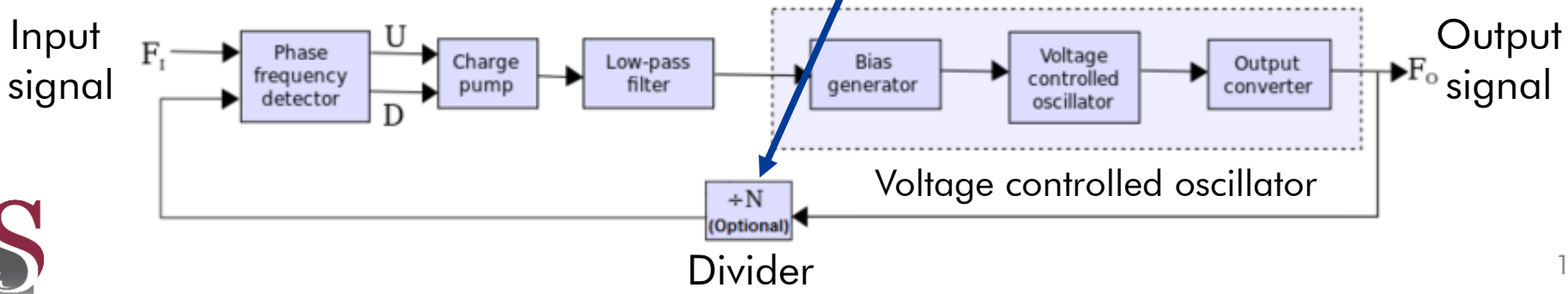


PLL and clock multiplication / Fase-sluit lus en klok vermenigvuldiging

- Use frequency divider (successive divide-by-2 counters) between PLL output and feedback – causes PLL to adjust to frequency that is N times higher than reference input.
- The output is changed so that frequency difference goes to 0.
- Using a single D flip-flop, divide by 2.
- Using two D flip-flops, divide by 4.







Clock distribution / Kloksein verspreiding

- Clocks are distributed to CPU, memory and peripherals
- Usually various selections (multiplexers) for clock input to different parts, including scaling through frequency dividers or multipliers
- Distribution of clocks is called the **clock tree.**
- Primary clock is the called the **system clock** (SYSCLK)
  - HSI RC oscillator clock (high speed internal) 16 MHz
  - HSE crystal oscillator clock (high speed external) 4 to 26 MHz
  - Main PLL (PLL) clock (scaled version of either HSI or HSE clock)
- Secondary clocks:
  - 32 kHz low-speed internal RC (LSI RC) which drives the independent watchdog and, optionally, the RTC used for Auto-wakeup from the Stop/Standby mode.
  - 32.768 kHz low-speed external crystal (LSE crystal) which optionally drives the RTC clock (RTCCLK)
- Each clock source can be switched on or off independently when it is not used, to optimize power consumption.

STM32F411 clocks (RC and PLL)

- On reset the 16 MHz internal RC oscillator is selected as the default CPU clock.
  - The 16 MHz internal RC oscillator is factory-trimmed to offer 1% accuracy at 25 °C.
- The application can then select as system clock either the RC oscillator or an external 4-26 MHz clock source.
  - This clock can be monitored for failure. If a failure is detected, the system automatically switches back to the internal RC oscillator and a software interrupt is generated (if enabled).
- This clock source is input to a PLL thus allowing to increase the frequency up to 100 MHz.
- Similarly, full interrupt management of the PLL clock entry is available when necessary (for example if an indirectly used external oscillator fails).

STM32F411 clocks (RC and PLL)

- Several prescalers allow the configuration of the two AHB buses, the high-speed **Advanced Peripheral Bus APB** (APB2) and the low-speed APB (APB1) domains.
- The maximum frequency of the two AHB buses and high-speed APB domains is 100 MHz.
- The maximum allowed frequency of the low-speed APB domain is 50 MHz.
- The devices embed a dedicated PLL (PLLI2S) which allows to achieve audio class performance.
  - In this case, the I2S master clock can generate all standard sampling frequencies from 8 kHz to 192 kHz.

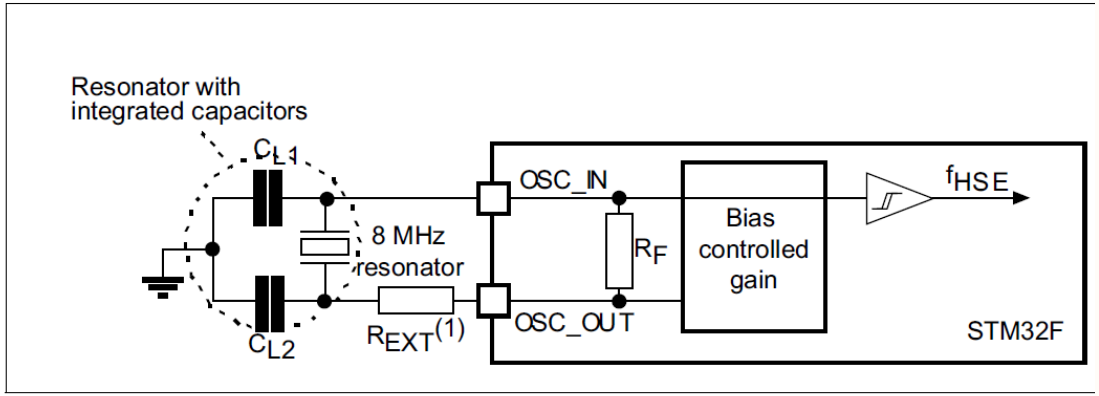
Real-time clock (RTC)

- The **real-time clock (RTC)** is an independent BCD timer/counter.
- Dedicated registers contain the second, minute, hour (in 12/24 hour), week day, date, month, year, in BCD (binarycoded decimal) format.
  - Correction for 28, 29 (leap year), 30, and 31 day of the month are performed automatically.
- The RTC provides a programmable alarm and programmable periodic interrupts with wakeup from Stop and Standby modes.
- The sub-seconds value is also available in binary format. It is clocked by a 32.768 kHz external crystal, resonator or oscillator, the internal low-power RC oscillator, with a typical frequency of 32 kHz, or the high-speed external clock divided by 128.
- It is by default configured to generate a time base of 1 second from a clock at 32.768 kHz.

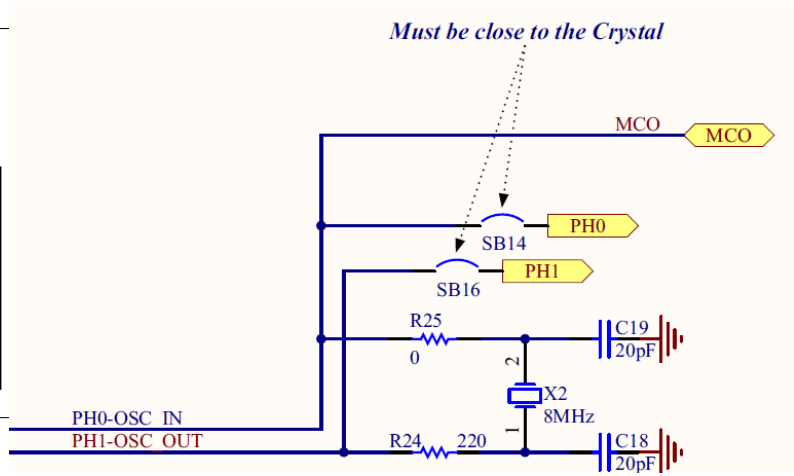
HSE for the STM32F

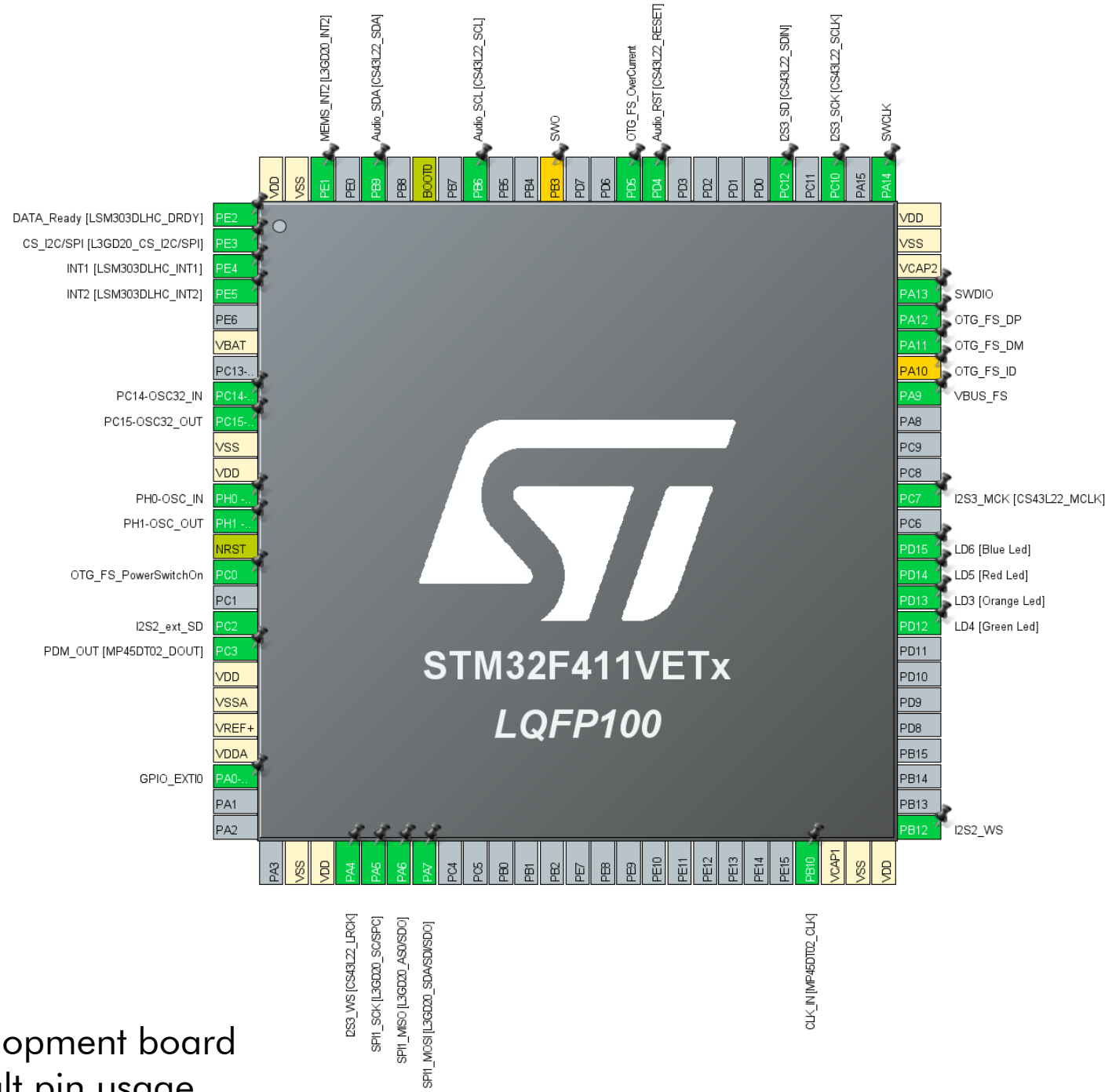
- For the HSE oscillator:
  - 4 to 26 MHz crystal/ceramic resonator oscillator
  - CL1 and CL2 use high-quality external ceramic capacitors in the 5 pF to 25 pF range, designed for high-frequency applications, and selected to match the requirements of the crystal or resonator. They should be the same value.
  - Connected to PH0 and PH1 = OSC_IN and OSC_OUT.

Figure 24. Typical application with an 8 MHz crystal



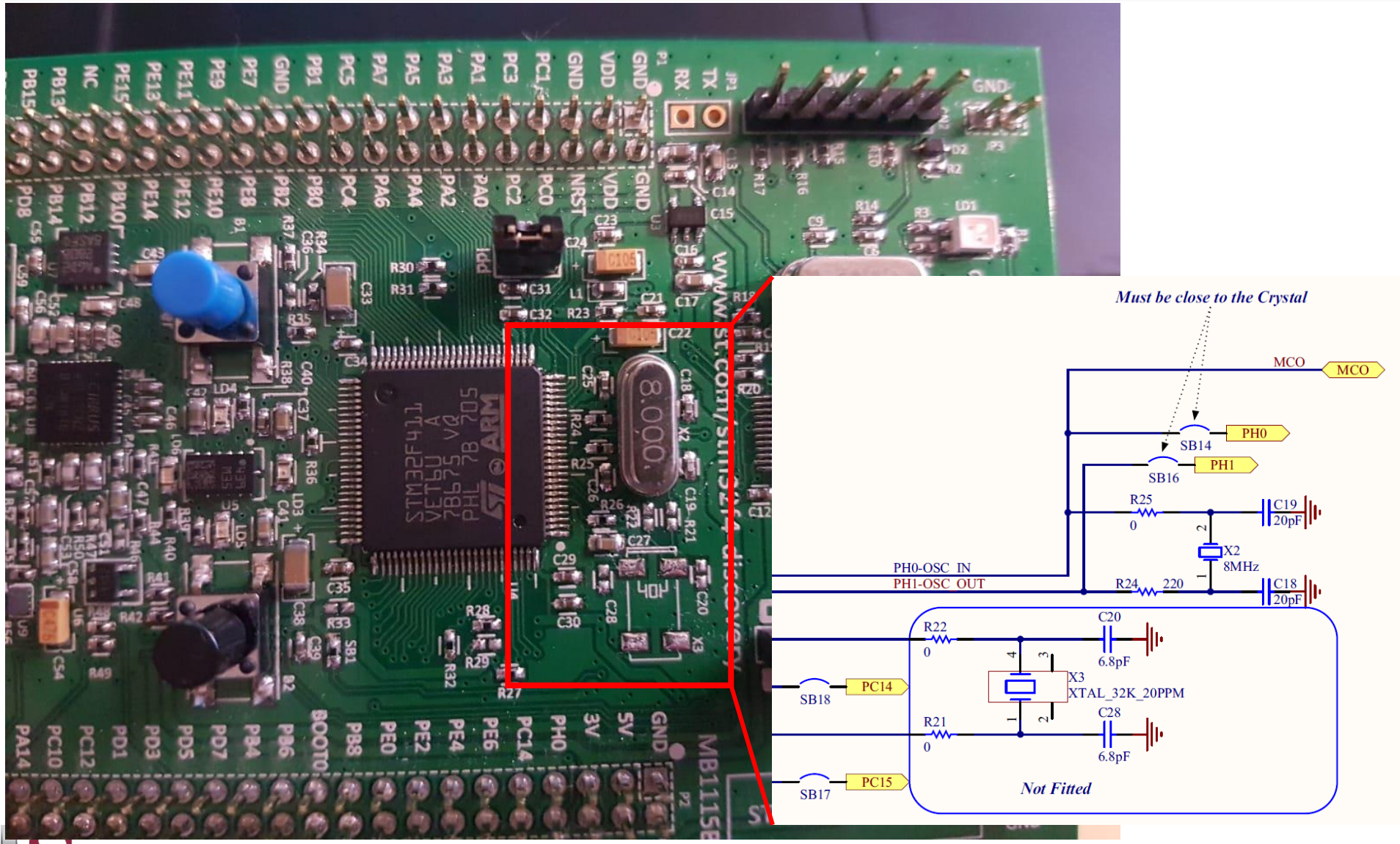
1. R_{EXT} value depends on the crystal characteristics.

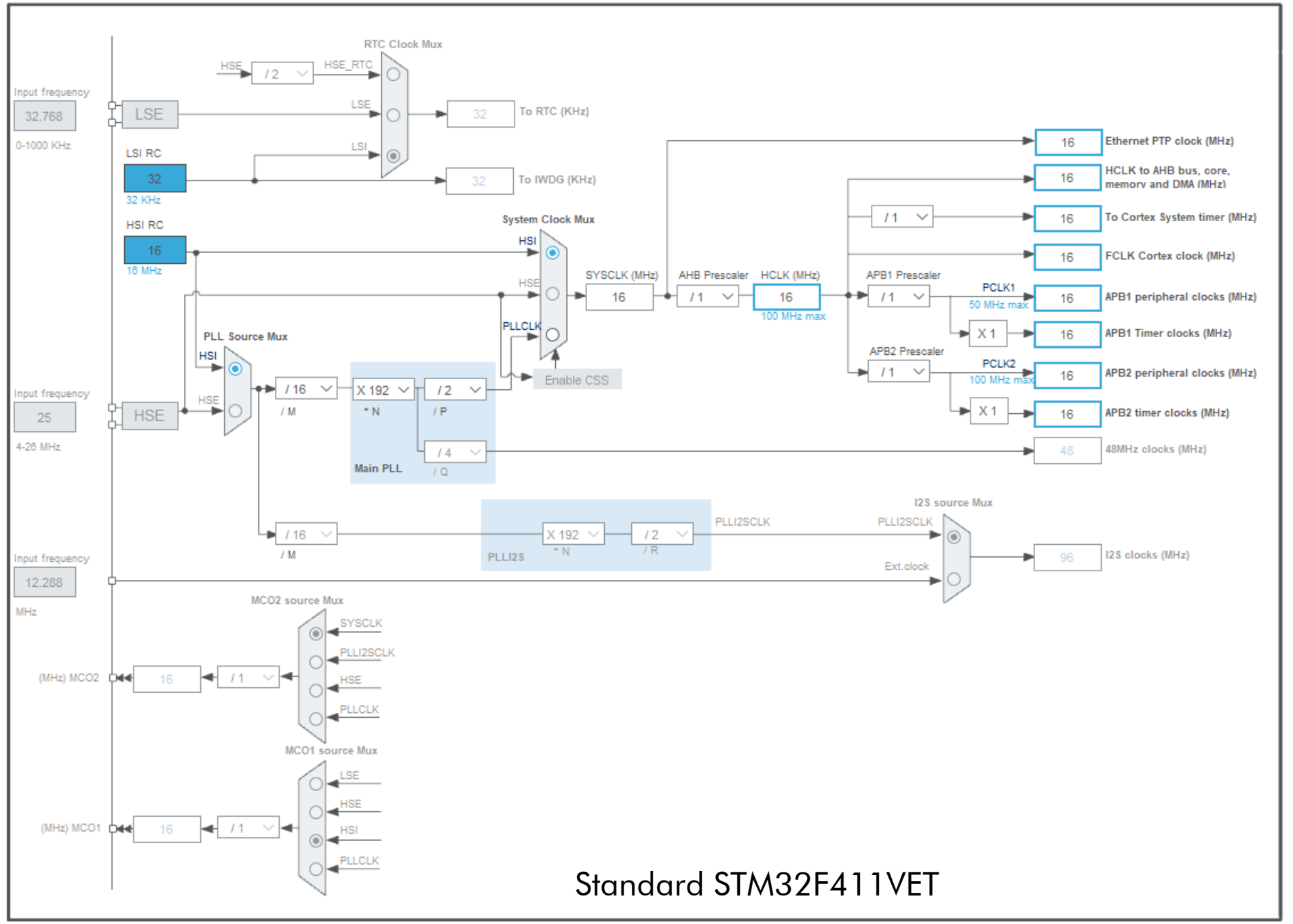




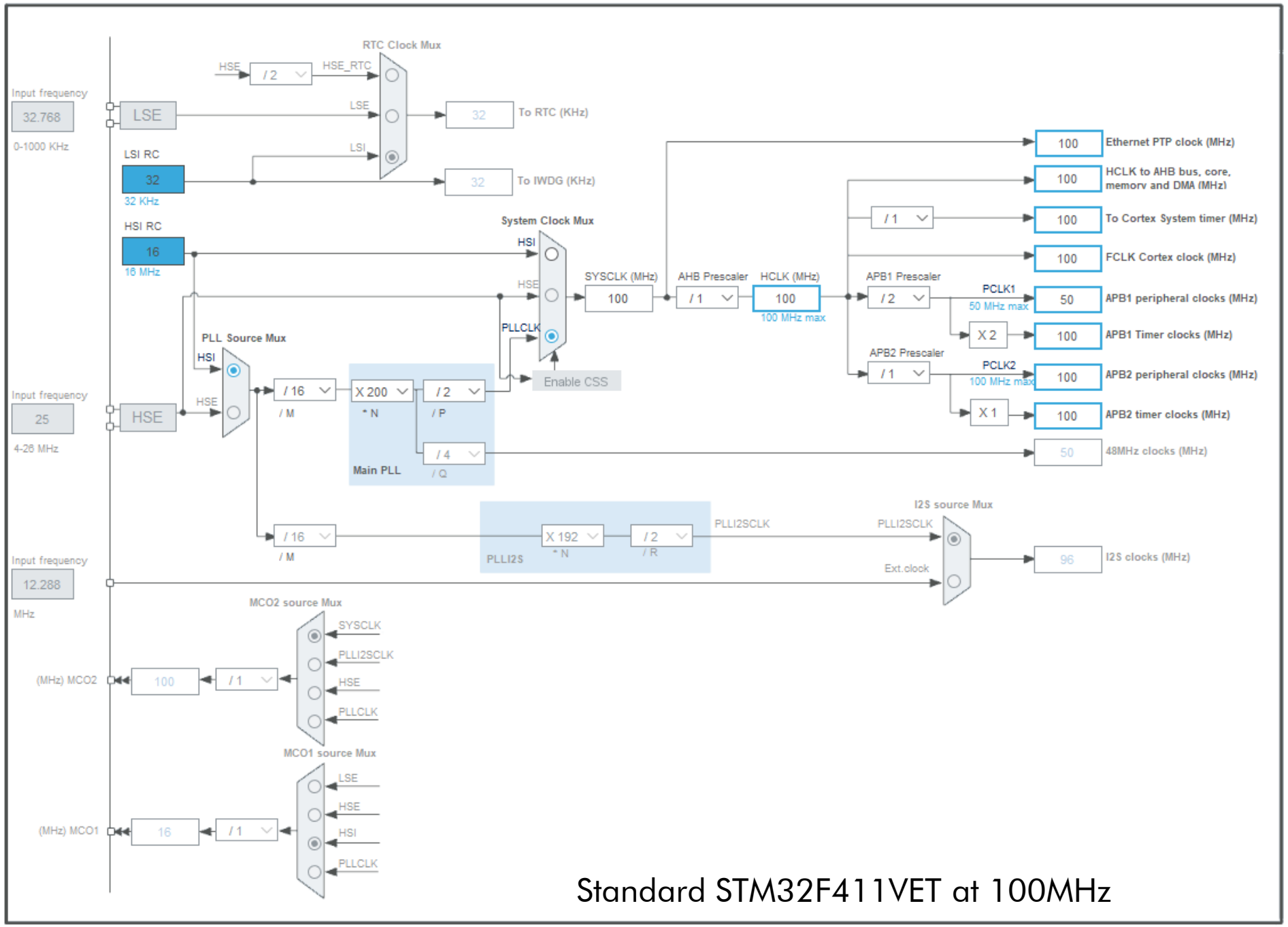
Development board default pin usage

HSE on the Development Board

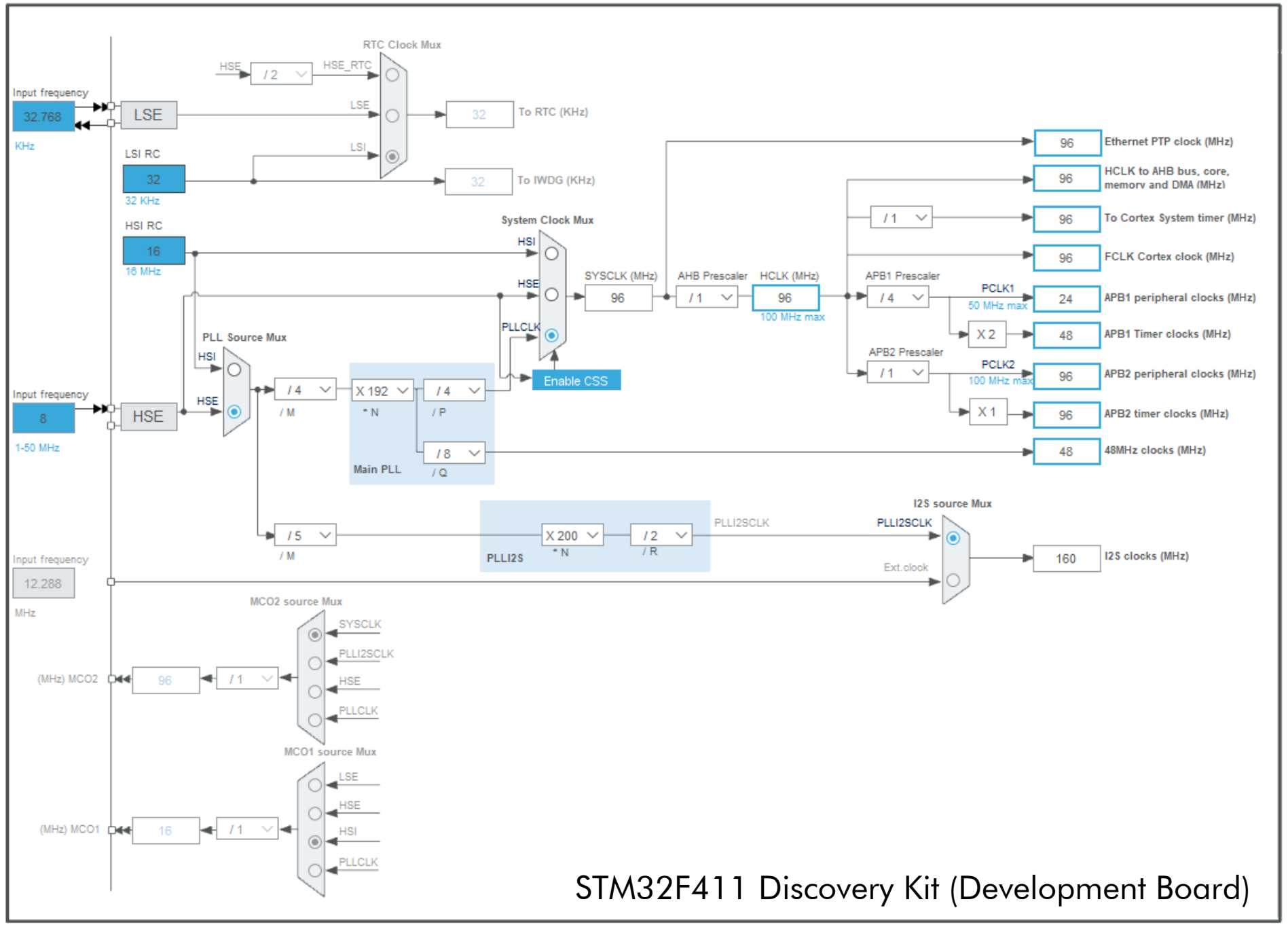




Standard STM32F411VET



Standard STM32F411VET at 100MHz



STM32F411 Discovery Kit (Development Board)

How do we setup the clock configuration for the STM32F411…?

- Clock configuration is controlled using memory mapped registers

6.3.1 RCC clock control register (RCC_CR)

Address offset: 0x00

Reset value: 0x0000 XX81 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
Reserved				PLLI2S RDY	PLLI2S ON	PLLRDY	PLLON	Reserved				CSS ON	HSE BYP	HSE RDY	HSE ON
				r	rw	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

How do we setup the clock configuration for the STM32F411…?

- PLL configuration register

6.3.2 RCC PLL configuration register (RCC_PLLCFGR)

Address offset: 0x04

Reset value: 0x2400 3010

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

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

- $f_{(VCO\ clock)} = f_{(PLL\ clock\ input)} \times (PLLN / PLLM)$
- $f_{(PLL\ general\ clock\ output)} = f_{(VCO\ clock)} / PLLP$
- $f_{(USB\ OTG\ FS,\ SDIO,\ RNG\ clock\ output)} = f_{(VCO\ clock)} / PLLQ$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				PLLQ3	PLLQ2	PLLQ1	PLLQ0	Reserved	PLLSRC	Reserved				PLLP1	PLLP0
				rw	rw	rw	rw		rw					rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PLLN									PLLM5	PLLM4	PLLM3	PLLM2	PLLM1	PLLM0
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- And more registers…

How to configure? Use STM32CubeIDE!

Clock speed record

CPU Frequency World Record ranking on 16 August 2020

RANK	SCORE	USER	PROCESSOR	COOLING	MOTHERBOARD
#1	8722.78 mhz	The Stilt	AMD FX-8370 @ 8722.8MHz	Liquid Nitrogen	ASUS ROG Crosshair V Formula-Z
#2	8709 mhz	AndreYang	AMD FX-8150 @ 8709MHz	Liquid Nitrogen	ASUS ROG Crosshair V Formula
#3	8659.64 mhz	Smoke	AMD FX-8370 @ 8659.6MHz	Liquid Nitrogen	ASUS ROG Crosshair V Formula-Z
#4	8615.39 mhz	slamms	AMD FX-8350 @ 8615.4MHz	Liquid Nitrogen	ASUS ROG Crosshair V Formula-Z
#5	8543.71 mhz	wytiwx	Intel Celeron D 352 @ 8543.7MHz	Liquid Nitrogen	ASUS P5E3 Premium/WiFi-AP @n
#6	8532.17 mhz	BenchBros	AMD FX-8320 @ 8532.2MHz	Liquid Nitrogen	ASUS ROG Crosshair V Formula-Z
#7	8520.22 mhz	CherV	AMD A10-6800K @ 8520.2MHz	Liquid Nitrogen	
#8	8502.1 mhz	NickShih	AMD FX-8350 @ 8502.1MHz	Liquid Nitrogen	
#9	8470.74 mhz	Hicookie	AMD FX-8350 @ 8470.7MHz	Liquid Nitrogen	GIGABYTE GA-990FXA-UD3
#10	8448.78 mhz	_12_	AMD FX-8320 @ 8448MHz	Liquid Nitrogen	
#11	8429.38 mhz	macci	AMD FX-8150 @ 8429.4MHz	Liquid Helium	ASUS ROG Crosshair V Formula
#12	8407.06 mhz	Atheros	AMD FX-8320 @ 8407.1MHz	Liquid Nitrogen	
#13	8406.34 mhz	Wizerty	AMD FX-8150 @ 8406.3MHz	Liquid Nitrogen	
#14	8370.93 mhz	alvinkenzo	AMD FX-8350 @ 8370.9MHz	Liquid Nitrogen	
#15	8366.83 mhz	The Silver	AMD FX-8370 @ 8366.8MHz	Liquid Nitrogen	ASUS ROG Crosshair V Formula-Z
#16	8348.43 mhz	Ananerbe	AMD FX-8350 @ 8348.4MHz	Liquid Nitrogen	
#17	8343.92 mhz	CtrlFix	AMD FX-9590 @ 8343.9MHz	Liquid Nitrogen	
#18	8337.98 mhz	der8auer	AMD FX-8350 @ 8338MHz	Liquid Nitrogen	ASUS ROG Crosshair V Formula
#19	8322.44 mhz	Power_VANO	Intel Celeron D 356 @ 8322.4MHz	Liquid Nitrogen	ASUS Maximus Formula
#20	8312.35 mhz	Niuulh	AMD FX-9590 @ 8312.4MHz	Liquid Nitrogen	ASUS ROG Crosshair V Formula

https://hwbot.org/benchmark/cpu_frequency/halloffame

Clock speed vs power consumption / Klok spoed en kragverbruik

- Power consumption increases as a result of faster switching between transistors.
- Higher clock speed = higher power consumption
- Higher power consumption = increase in temperature
- On modern microcontrollers, peripherals can be disabled selectively (i.e. switch off the clock signal going to the I/O port), and various low-power states and wake-up capabilities.