Microprocessors

2nd lecture: STM32 intro, GPIO

Brief ARM History

- **A**corn **R**ISC **M**achine. Acorn was a computer company in the UK in the 1980s
  - Headquarters in Cambridge, relatively small company
- Wanted a chip to succeed 6502. Decided to make one themselves.
  - 6502 was the chip in Commodore 64, Apple II, NES, Atari 2600

ARM Business Plan

- IP Licensing company. Does not manufacture own chips.
- Other companies take the design, put on SoC, attach whatever other logic blocks are needed
  - Alcatel-Lucent, Apple Inc., Atmel, Broadcom, Cirrus Logic, Digital Equipment Corporation, Freescale, DEC, LG, Marvell, Microsoft, NEC, Nintendo, Nvidia, Sony, NXP, Oki, ON Semiconductor, Qualcomm, Samsung, Sharp, STMicroelectronics, Texas Instruments, VLSI Technology, Yamaha and more

AMBA Bus Protocol

Advanced Microcontroller Bus Architecture

- ARM System Bus (ASB), ARM Peripheral Bus (APB)
- ARM High Performance Bus (AHB)
- Common bus, various companies can provide logic blocks for it, can swap in and out ARM cores as needed.

Models -- Confusing

Architecture vs Family

- ARMv1 : ARM1
- ARMv2 : ARM2, ARM3 (26-bit, status in PC register)
- ARMv3 : ARM6, ARM7
- ARMv4 : StrongARM, ARM7TDMI, ARM9TDMI
- ARMv5 : ARM7EJ, ARM9E, ARM10E, XScale
- ARMv6 : ARM11, ARM Cortex-M0 (Raspberry Pi)
- ARMv7 : Cortex A8, A9 ,Cortex-M3 (iPad, iPhone, Pandaboard, Beagleboard, Beaglebone)
- ARMv8 : Cortex A-50 (64-bit)

Various abbreviations in Model Names

For example, ARM7DTMI

- "Application" ARM Cortex-A
- "Real-time" ARM Cortex-R
- "Micro-controller" ARM Cortex-M
- "E" means DSP instructions
- "M" improved multiplier
- "T" THUMB
- "J" Jazelle (java bytecodes)
- D/I Debug/ICE
- EE -- ThumbExecutionEnvironment, Just-in-time
- NEON -- SIMD
- VFP -- Floating point

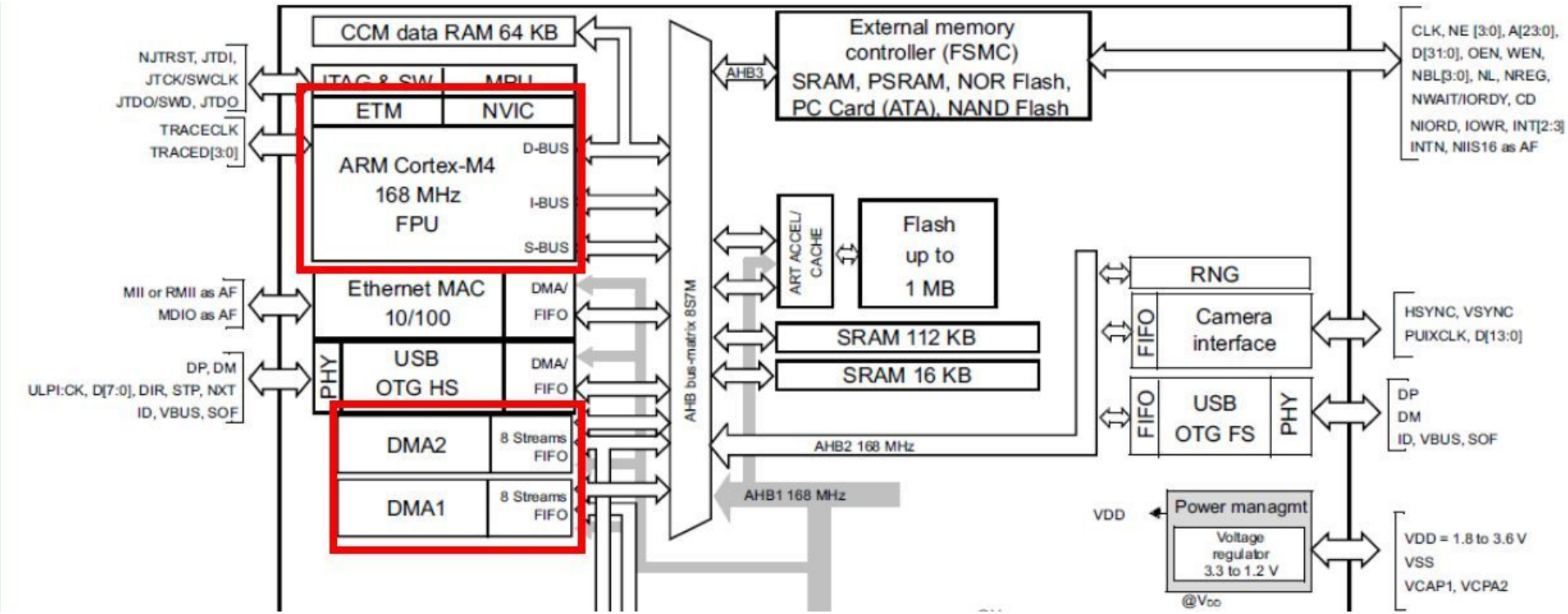
STM32F4 1/2

- ARM Cortex-M4F core, 180 MHz. F is for Floating point
- Static RAM, 64K core coupled memory (CCM), 4K battery-backed, 80B tamper-detect erase.
- Flash ROM: 512 - 2048 KB general purpose, 30 KB system boot
- Lots of busses: USB, CAN, SPI, I2S, I2C, UART, SDIO for SD/MMC, ADCs, DACs, GPIOs, DMA, RTC, CRC engine, RNG
- Some packages support external memory bus
- Instruction set: Thumb, Thumb-2, Saturating Math, DSP, FPU
- ARMv7E-M architecture

STM32F4 2/2

- 1-cycle 32-bit hardware multiply, 2-12 cycle 32-bit hardware divide, saturated math support
- DSP extension: Single cycle 16/32-bit MAC, single cycle dual 16-bit MAC, 8/16-bit SIMD arithmetic.
- Floating-Point extension (silicon option): Single-precision floating point unit, IEEE-754 compliant.
- 3-stage pipeline with branch speculation
- optional 8 region memory protection unit (MPU)

STM32F4 architecture



STM32F4 vs older families

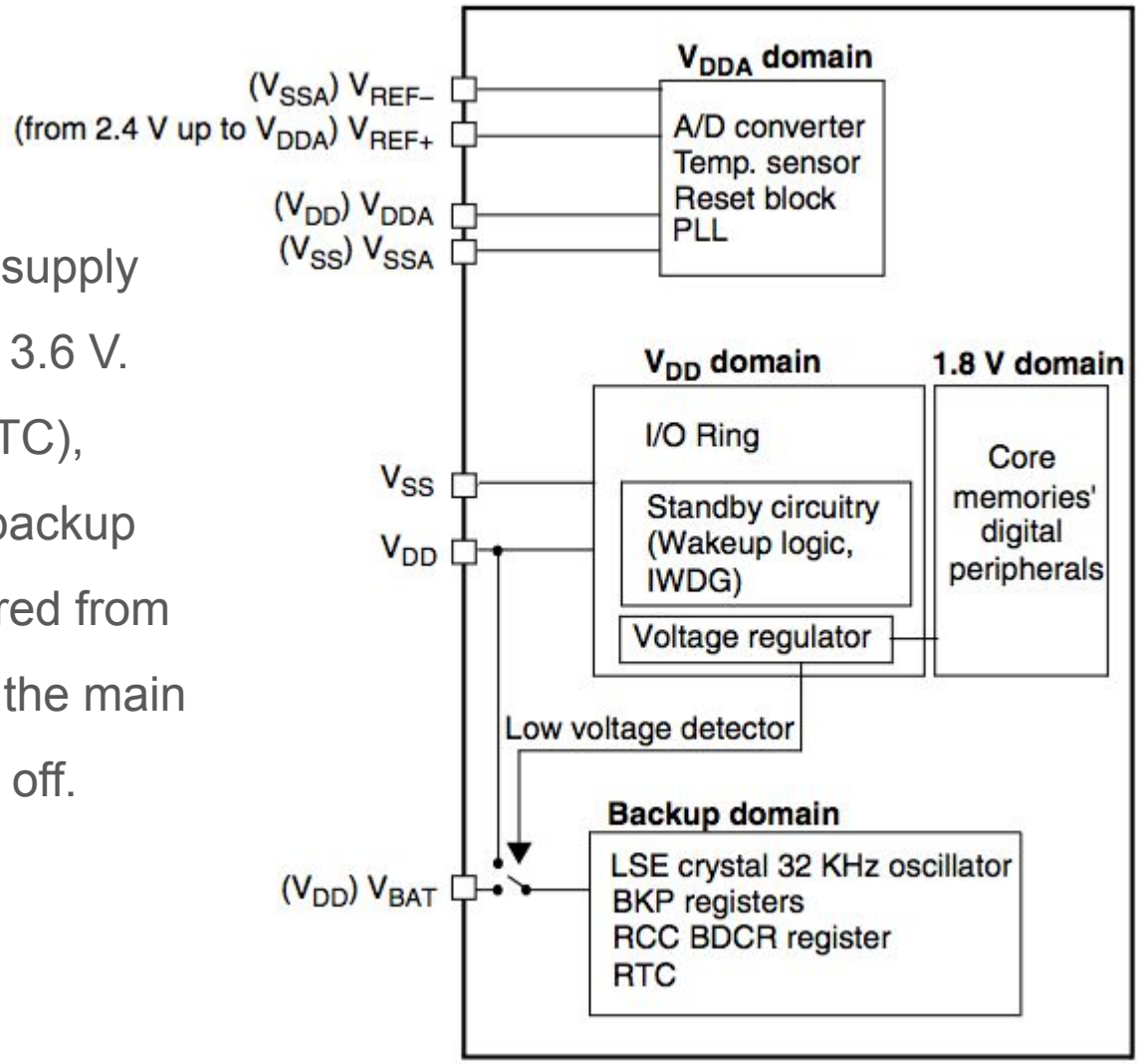
Peripherals	STM32 F1 series	STM32 F2 series	STM32 F4 series
The need for speed			
USB FS	12 Mbit/s	12 Mbit/s	12 Mbit/s
USB HS	-	480 Mbit/s	480 Mbit/s
USART		Up to 7.5 Mbit/s	Up to 10.5 Mbit/s
SPI	Up to 18 Mbit/s	Up to 30 Mbit/s	Up to 37.5 Mbit/s
I²C	400 kHz	400 kHz	400 kHz
GPIO	Up to 18 MHz	Up to 60 MHz	Up to 60 MHz
3-phase MC timer	72 MHz PWM timer clock input	120 MHz PWM timer clock input	168 MHz PWM timer clock input
SDIO	Up to 48 MHz	Up to 48 MHz	Up to 48 MHz
I²S	From 8 kHz to 96 kHz sampling frequencies	From 8 kHz to 96 kHz sampling frequencies	From 8 kHz to 96 kHz sampling frequencies
Camera interface	-	Up to 48 Mbyte/s at 48 MHz	Up to 67.2 Mbyte/s at 67.2 MHz
Crypto/hash processor	-	AES-256 up to 106 Mbyte/s	AES-256 up to 149.33 Mbyte/s
FSMC	Up to 36 MHz	Up to 60 MHz	Up to 60 MHz
The need for analog			
ADC conversion time	1 µs (1 MSPS)	0.5 µs (2 MSPS)	0.41 µs (2.44 MSPS)
DAC	2-channel, 12-bit	2-channel, 12-bit	2-channel, 12-bit
The need for connectivity			
CAN	Up to 2 independent CAN	Up to 2 independent CAN	Up to 2 independent CAN
Ethernet	10/100 Mbit/s MAC with hardware IEEE 1588	10/100 Mbit/s MAC with hardware IEEE 1588	10/100 Mbit/s MAC with hardware IEEE 1588
USB OTG	Full speed host, device or OTG	Full speed and high speed host, device or OTG	Full speed and high speed host, device or OTG
CEC bus	Consumer electronics control for consumer devices	-	-
Flexible static memory	4 independent banks, 8/16-bit data bus, supports SRAM, PSRAM, NAND and NOR Flash, parallel graphic LCD	4 independent banks, 8/16-bit data bus, supports SRAM, PSRAM, NAND and NOR Flash, parallel graphic LCD	4 independent banks, 8/16-bit data bus, supports SRAM, PSRAM, NAND and NOR Flash, parallel graphic LCD
Camera interface	8- to 14-bit parallel	8- to 14-bit parallel	8- to 14-bit parallel

STM32F4

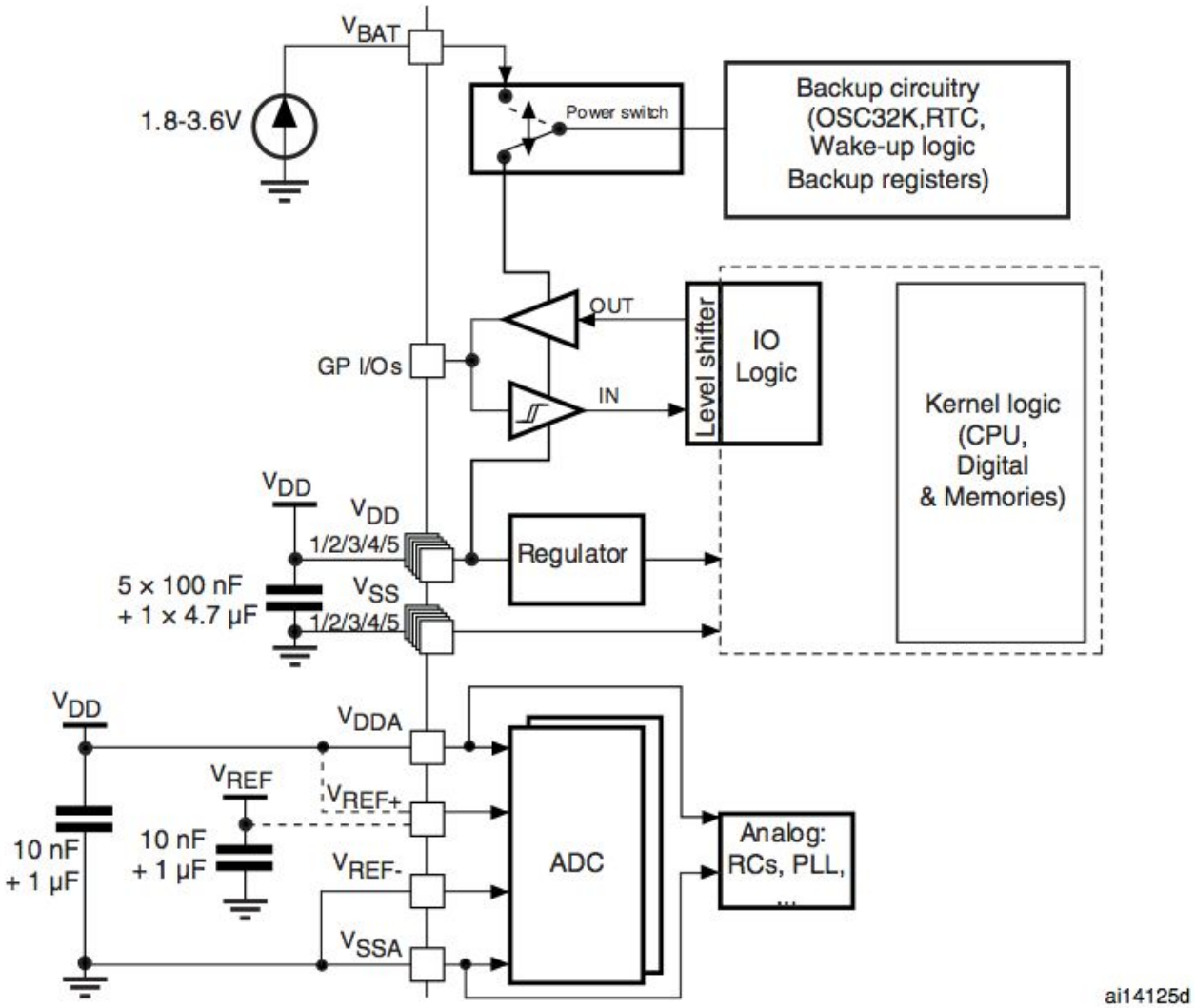
1. **Hardware overview**
    a. Power supply
    b. Reset circuit
    c. Boot mode selection
    d. Clock source
    e. Debug interface
2. **Clock configuration**
    a. STM32 clock tree
    b. RCC registers
    c. RCC registers
3. **GPIO peripheral**
    a. GPIO overview
    b. GPIO registers

Power supply (1/2)

- The operating voltage supply (V_{DD}) range is 1.8 V to 3.6 V.
- The real-time clock (RTC), backup registers and backup registers can be powered from the V_{BAT} voltage when the main V_{DD} supply is powered off.

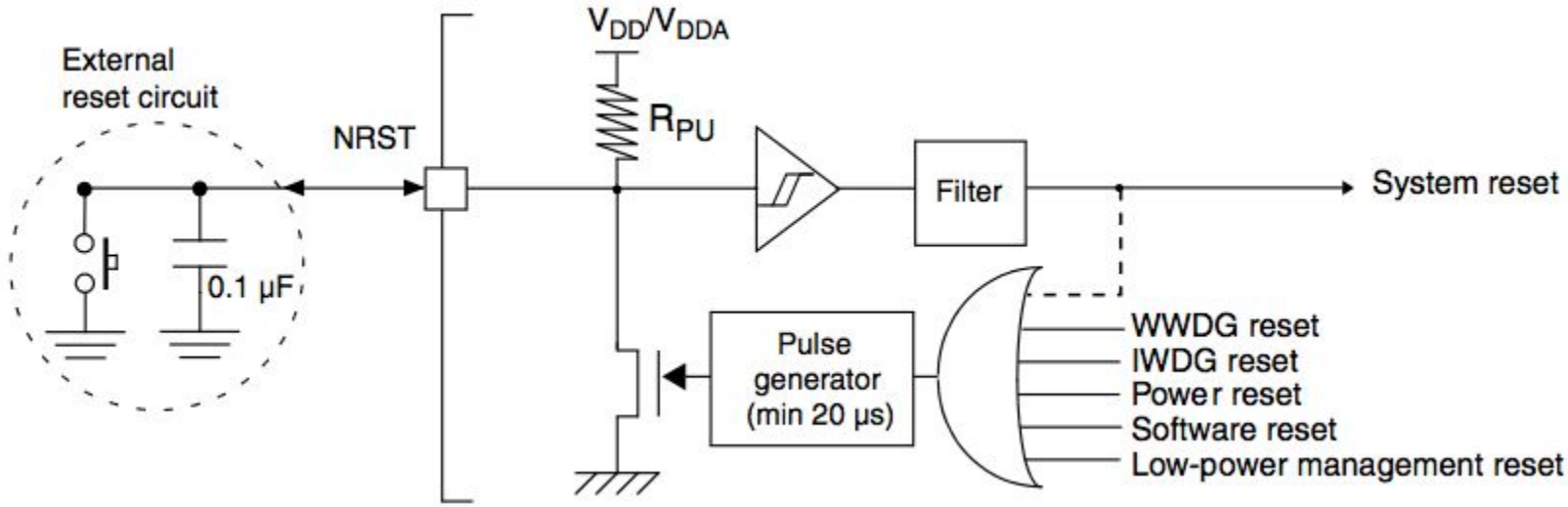


Power supply (2/2)



Reset circuit

- The STM32F4xx does not require an external reset circuit to power-up correctly
- A pull-down capacitor is recommended to improve EMS performance by protecting the device against parasitic resets
- The capacitor recommended value (100 nF) can be reduced to 10 nF to limit this power consumption



System reset

A system reset sets all registers to their reset values except for the reset flags in the clock controller CSR register and the registers in the Backup domain (slide 11)

A system reset is generated when one of the following events occurs:

1. A low level on the NRST pin (external reset)
2. Window watchdog end-of-count condition (WWDG reset)
3. Independent watchdog end-of-count condition (IWDG reset)
4. A software reset (SW reset)
5. Low-power management reset

The reset source can be identified by checking the reset flags in the Control/Status register, RCC_CSR.

Real-time clock (RTC) and backup registers

The backup domain includes:

- The real-time clock (RTC)
  - Independent BCD timer/counter
  - The RTC features a reference clock detection, a more precise second source clock (50 or 60 Hz) can be used to enhance the calendar precision.
  - The RTC provides a programmable alarm and programmable periodic interrupts with wakeup from Stop and Standby modes
  - It is clocked by a 32.768 kHz external crystal, resonator or oscillator, the internal low-power RC oscillator or the high-speed external clock divided by 128
  - Two alarm registers are used to generate an alarm at a specific time and calendar fields can be independently masked for alarm comparison. To generate a periodic interrupt, a 16-bit programmable binary auto-reload downcounter with programmable resolution is available and allows automatic wakeup and periodic alarms from every 120 μs to every 36 hours.
- 20 backup registers
  - 32-bit registers used to store 80 bytes of user application data when V_{DD} power is not present.
  - Backup registers are not reset by a system, a power reset, or when the device wakes up from the Standby mode

Low-power modes

The devices support three low-power modes to achieve the best compromise between low power consumption, short startup time and available wakeup sources:

- Sleep mode
  - In Sleep mode, only the CPU is stopped.
  - All peripherals continue to operate and can wake up the CPU when an interrupt/event occurs.
- Stop mode
  - The Stop mode achieves the lowest power consumption while retaining the contents of SRAM and registers.
  - All clocks in the 1.2 V domain are stopped, the PLL, the HSI RC and the HSE crystal oscillators are disabled.
  - The voltage regulator can also be put either in normal or in low-power mode.
  - The device can be woken up from the Stop mode by any of the EXTI line (the EXTI line source can be one of the 16 external lines, the PVD output, the RTC alarm/ wakeup/tamper/ time stamp events).

Low-power modes

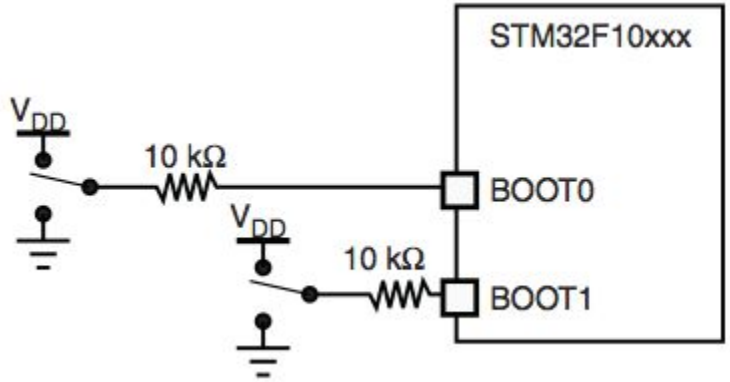
- Standby mode
  - The Standby mode is used to achieve the lowest power consumption.
  - The internal voltage regulator is switched off so that the entire 1.2 V domain is powered off.
  - The PLL, the HSI RC and the HSE crystal oscillators are also switched off
  - After entering Standby mode, the SRAM and register contents are lost except for registers in the backup domain when selected.

The device exits the Standby mode when an external reset (NRST pin), an IWDG reset, a rising edge on the WKUP pin, or an RTC alarm/ wakeup/ tamper/ timestamp event occurs.

Peripheral power consumption

Peripheral		I_{DD} (typ)	Unit
AHB1 (up to 84MHz)	GPIOA	1.55	µA/MHz
	GPIOB	1.55	
	GPIOC	1.55	
	GPIOD	1.55	
	GPIOE	1.55	
	GPIOH	1.55	
	CRC	0.36	
	DMA1	20.24	
	DMA2	21.07	
APB1 (up to 42MHz)	TIM2	11.19	µA/MHz
	TIM3	8.57	
	TIM4	8.33	
	TIM5	11.19	
	PWR	0.71	
	USART2	3.33	
	I2C1/2/3	3.10	
	SPI2(1)	2.62	
	SPI3(1)	2.86	
	I2S2	1.90	
	I2S3	1.67	
	WWDG	0.71	

Boot mode



BOOT mode selection pins		Boot mode	Aliasing
BOOT1	BOOT0		
x	0	Main Flash memory	Main Flash memory is selected as boot space
0	1	System memory	System memory is selected as boot space
1	1	Embedded SRAM	Embedded SRAM is selected as boot space

The values on the BOOT pins are latched on the 4th rising edge of SYSCLK after a reset. It is up to the user to set the BOOT1 and BOOT0 pins after reset to select the required boot mode.

Boot mode

Boot from User Flash mode

- The application code that runs after reset is located in user flash memory.
- The user flash memory in this mode is aliased to start at address 0x00000000 in boot memory space.
- Upon reset, the top-of-stack value is fetched from address 0x00000000, and code then begins execution at address 0x00000004.

Boot from System Memory mode

- The system memory (not the user flash) is now aliased to start at address 0x00000000.
- The application code in this case must have already been loaded into system memory.

Boot mode

Boot from Embedded SRAM mode

- The SRAM start at address 0x00000000.
- When this mode is selected, the device expects the vector table to have been relocated using the NVIC exception table and offset register, and execution begins at the start of embedded SRAM.
- The application code in this case must have already been loaded into embedded SRAM.
- This last mode is usually used for Debugging.

Embedded boot loader

0	1	System memory	System memory is selected as boot space

The Embedded boot loader mode is used to reprogram the Flash memory using one of the available serial interfaces:

- in low-density, low-density value line, medium-density, medium-density value line, and high-density devices the boot loader is activated through the USART1 interface (AN2606)
- In connectivity line devices the boot loader can be activated through USART1, USART2 (remapped), CAN2 (remapped) or USB OTG FS in device firmware upgrade) mode (AN2662)

STM32F4 clock sources

Three different clock sources can be used to drive the main system clock:

- HSI oscillator clock (high-speed internal clock signal)
- HSE oscillator clock (high-speed external clock signal)
- PLL clock

The devices have two secondary clock sources:

- 32 kHz low-speed internal RC (LSI RC) that drives the independent watchdog
- 32.768 kHz low-speed external crystal (LSE crystal) that optionally drives the real-time clock (RTCCLK)

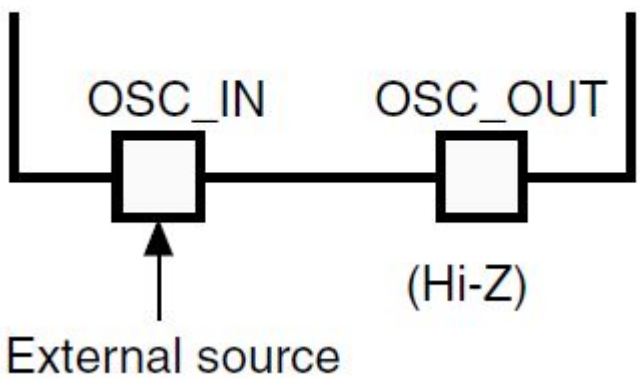
Each clock source can be switched on or off independently when it is not used, to optimize the power consumption.

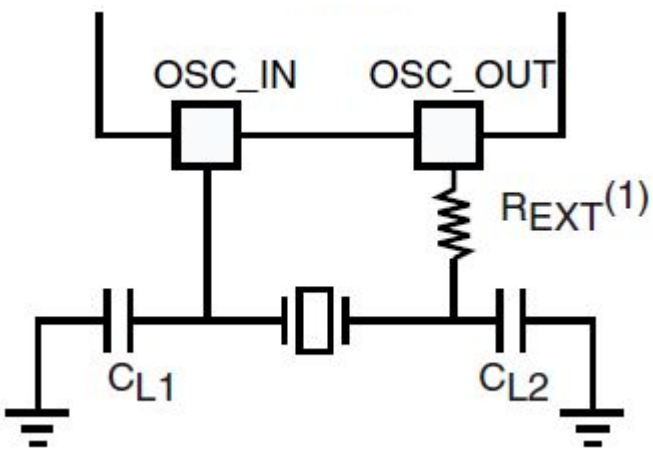
HSE clock

The high-speed external clock signal (HSE) can be generated from two possible clock sources:

- HSE user external clock
- HSE external crystal/ceramic resonator (16Mhz)

The value of R_{EXT} depends on crystal and oscillator port characteristics, usually it is about 1 kΩ

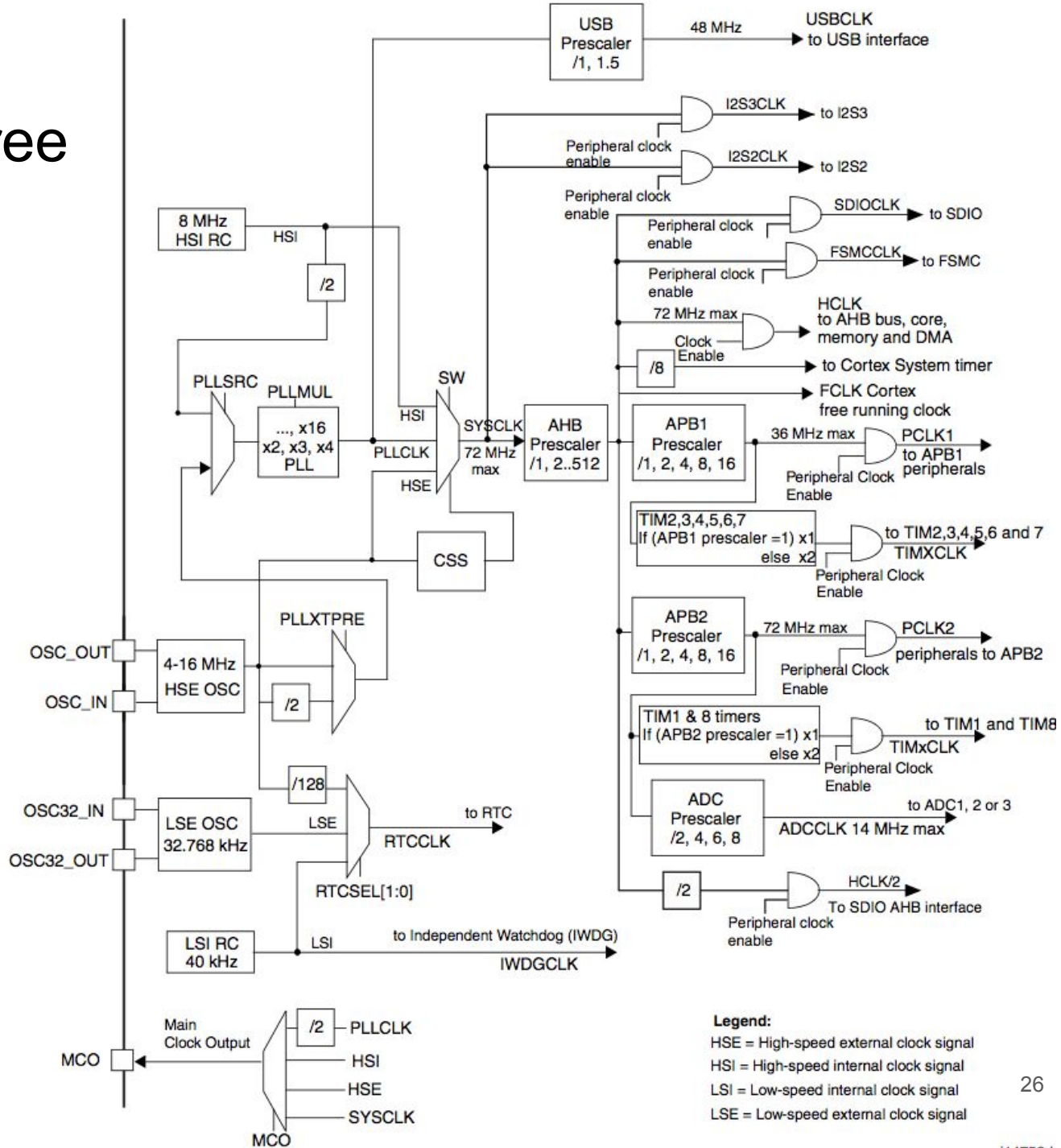




Clock precision and power consumption

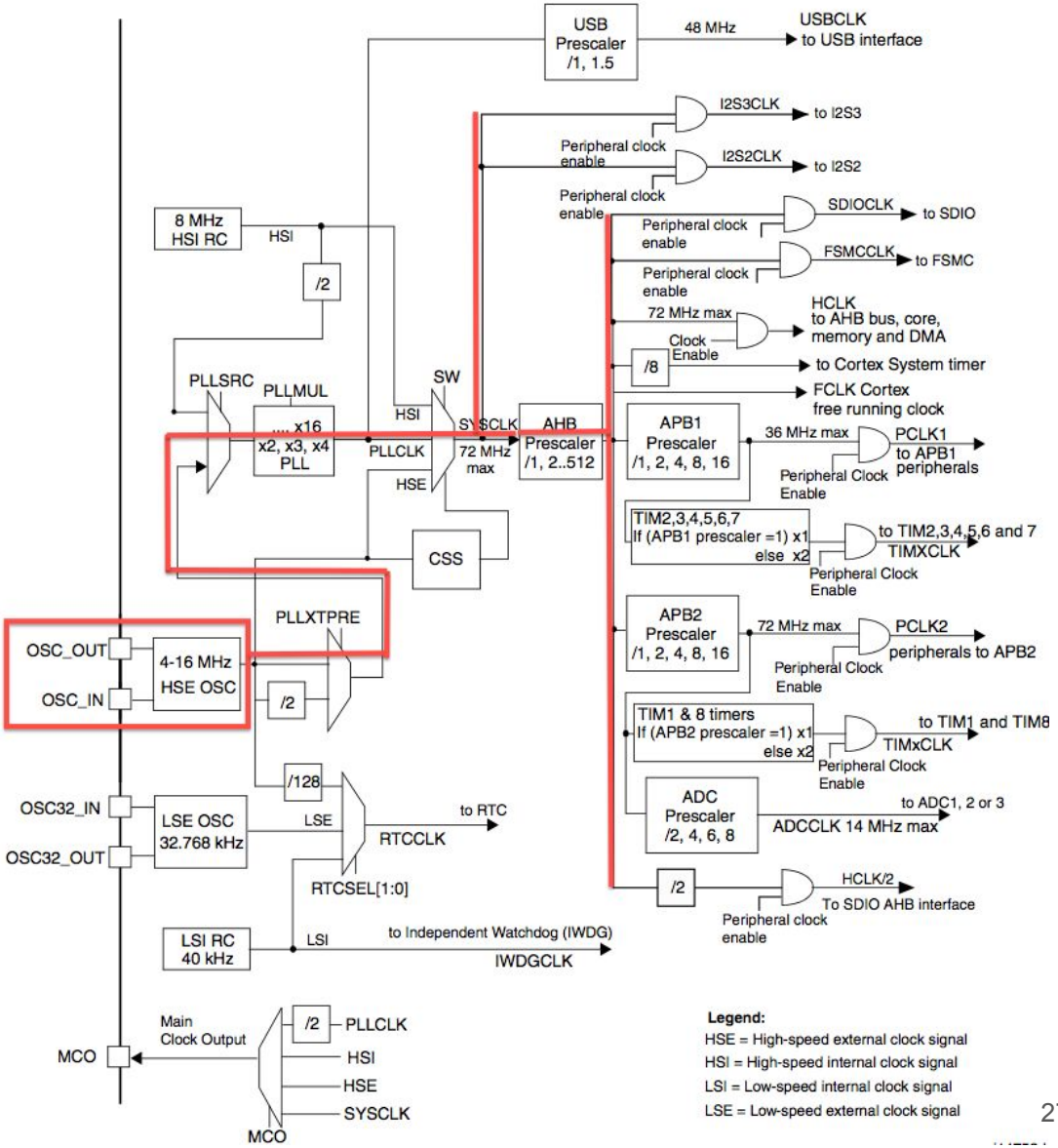
Osc/Clocks	Speed	Cons.	Precision 25°C/0-85°C	Wakeup
HSE ext crystal	1-24 MHz	~500 µA	~0.01% (100ppm)	1ms
HSE ext clock	1-32 MHz		NA	NA
HSI	16 MHz	100 µA	1%/2.5%	3.7 µs
MSI	65 kHz-4.2 MHz	0.7-15 µA	0.5%/3%	3.7 µs
PLL	2-32 MHz	350 µA	NA	100 µs
LSI	37 kHz	0.4 µA	50%	200 µs
LSE ext crystal	32.7 kHz typ	0.5 µA	~0.002%(20ppm)	~1s
LSE ext clock	1-1000 kHz		NA	NA

STM32 clock tree



PLL clock generator with HSE source

- 8 Mhz HSE input
- 72Mhz SYSCLK target
- PLL divider / multiplier



Clock configuration

RCC - Reset and Clock Control - clocks for buses / peripheries

- AHB1 – USB OTG, Ethernet MAC, DMA2, DMA1, CRC module, GPIOx
- AHB2 – USB OTG, Random generátor, Hash module, Cryptographic module, Camera interface
- AHB3 – Flexible static memory controller
- APB1 – DAC, Power interface, CAN2, CAN1, I2C2, I2C1, UART 2-5, SPI2, SPI3, Watchdog, TIM 2-7 a 12-14
- APB2 – TIM 9-11, System configuration controller, SPI1, SDIO, ADC, USART 1 a 6, TIM 8 a 1

RCC definition: RCC_TypeDef (header file stm32f4x.h)

Proper clock definition: system_STM32F4x.h

How to setup clocks?

6.3.1 Clock control register (RCC_CR)

Address offset: 0x00

Reset value: 0x0000 XX83 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						PLL RDY	PLLON	Reserved				CSS ON	HSE BYP	HSE RDY	HSE ON
						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

Bits 31:26 Reserved, always read as 0.

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

Set by hardware to indicate that the PLL is locked.
0: PLL unlocked
1: PLL locked

Bit 24 **PLLON:** PLL enable

Set and cleared by software to enable PLL.
Cleared by hardware when entering Stop or Standby mode. This bit can not be reset if the PLL clock is used as system clock or is selected to become the system clock.
0: PLL OFF
1: PLL ON

Clock configuration register (RCC_CFGR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						PLL RDY	PLLON	Reserved				CSS ON	HSE BYP	HSE RDY	HSE ON
						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

- **HSEON**: External high-speed clock enable - set and cleared by software
  - 0: HSE oscillator OFF
  - 1: HSE oscillator ON
- **HSERDY**: External high-speed clock ready flag set by hardware to indicate that the external oscillator is stable
  - 0: external oscillator not ready
  - 1: external oscillator ready
- **PLLON**: PLL enable
- **PLLRDY**: PLL clock ready flag

Clock control register (RCC_CR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					MCO[2:0]			Res.	USB PRE	PLLMUL[3:0]				PLL XTPRE	PLL SRC
					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
ADC PRE[1:0]		PPRE2[2:0]			PPRE1[2:0]			HPRE[3:0]				SWS[1:0]		SW[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	rw	rw

- **SW**: System clock switch - set and cleared to select SYSCLK source.
  - 00: HSI selected as system clock
  - 01: HSE selected as system clock
  - 10: PLL selected as system clock
  - 11: not allowed
- **HPRE**: AHB prescaler
- **PPRE1**: APB low-speed prescaler (APB1)
- **PPRE2**: APB high-speed prescaler (APB2)
- **PLLXTPRE**: HSE divider for PLL entry
  - 0: HSE clock not divided
- **PLLMUL**: PLL multiplication factor
  - 0111: PLL input clock x 9

Peripheral clock enable register (RCC_AHB1ENR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reser-ved	OTGHS ULPIEN	OTGHS EN	ETHMA CPTPE N	ETHMA CRXEN	ETHMA CTXEN	ETHMA CEN	Reserved		DMA2EN	DMA1EN	CCMDATA RAMEN	Res.	BKPSR AMEN	Reserved	
	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
Reserved			CRCEN	Reserved			GPIOIE N	GPIOH EN	GPIOGE N	GPIOFE N	GPIOEEN	GPIOD EN	GPIOC EN	GPIOB EN	GPIOA EN
			rw				rw	rw	rw	rw	rw	rw	rw	rw	rw

- GPIOAEN: IO port A clock enable - set and cleared by software.
  - 0: IO port A clock disabled
  - 1: IO port A clock enabled
- GPIOBEN: IO port B clock enable - set and cleared by software.
  - 0: IO port B clock disabled
  - 1: IO port B clock enabled
- GPIOCEN: IO port C clock enable - set and cleared by software.
  - 0: IO port C clock disabled
  - 1: IO port C clock enabled
- …

STM32F4 memory map

Bus	Boundary address	Peripheral
	0xE010 0000 - 0xFFFF FFFF	Reserved
Cortex®-M4	0xE000 0000 - 0xE00F FFFF	Cortex-M4 internal peripherals
	0x5004 0000 - 0xDFFF FFFF	Reserved
AHB2	0x5000 0000 - 0x5003 FFFF	USB OTG FS
AHB1	0x4002 6800 - 0x4FFF FFFF	Reserved
	0x4002 6400 - 0x4002 67FF	DMA2
	0x4002 6000 - 0x4002 63FF	DMA1
	0x4002 5000 - 0x4002 4FFF	Reserved
	0x4002 3C00 - 0x4002 3FFF	Flash interface register
	0x4002 3800 - 0x4002 3BFF	RCC
	0x4002 3400 - 0x4002 37FF	Reserved
	0x4002 3000 - 0x4002 33FF	CRC
	0x4002 2000 - 0x4002 2FFF	Reserved
	0x4002 1C00 - 0x4002 1FFF	GPIOH
	0x4002 1400 - 0x4002 1BFF	Reserved
	0x4002 1000 - 0x4002 13FF	GPIOE
	0x4002 0C00 - 0x4002 0FFF	GPIOD
	0x4002 0800 - 0x4002 0BFF	GPIOC
	0x4002 0400 - 0x4002 07FF	GPIOB
	0x4002 0000 - 0x4002 03FF	GPIOA

General Purpose I/O Port

- Each of the GPIO pins can be configured by software as output (push-pull or open-drain, with or without pull-up or pull-down), as input (floating, with or without pull-up or pull-down) or as peripheral alternate function.
- Each pin can simultaneously be configured as one of 16 external interrupt lines.
- Most of the GPIO pins are shared with digital or analog alternate functions.
- All GPIOs are high-current-capable and have speed selection to better manage internal noise, power consumption and electromagnetic emission.
- The I/O configuration can be locked if needed by following a specific sequence in order to avoid spurious writing to the I/Os registers.
- Fast I/O handling allowing maximum I/O toggling up to 84 MHz.

General Purpose I/O Port

- GPIO ports are named A-E and are all 5V tolerant.
- Many of the external pins may be switched from general purpose IO to serve as the Input/Output of a user peripheral, for example a USART or I2C peripheral.
- Additionally there is an external interrupt unit which allows 16 external interrupt lines to be mapped onto any combination of GPIO lines.
- It has registers to write word-wide or for atomic bit manipulation.
- Once a configuration is defined it can be locked.

GPIO port mode register (GPIOx_MODER)

This is a 32-bit register where each set of two consecutive bits represent the mode of a single I/O pin. For example bits 0 and 1 of the MODER register associated with GPIOC (GPIOC_MODER), represent the mode of GPIO pin PC0 and bits 26 and 27 of the same register represent the mode of GPIO pin PC13. These two bits can be set to:

- '00'-> input mode, which allows the GPIO pin to be used as an input pin,
- '01'-> Output mode, which allows the GPIO pin to be used as an output pin,
- '11'-> Analog mode, which allows the GPIO pin to be used as an Analog input
- '10'-> Alternate function mode which allow the GPIO pins to be used by peripherals such as the UART, SPI e.t.c. It is important to note that if a pin's MODE is set to alternate function, any GPIO settings for that pin in the GPIO registers will be overridden by the peripheral. I will be addressing Alternate function mode in more detail in a future entry.

GPIO port mode register (GPIOx_MODER)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	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
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 2y+1:2y **MODERy[1:0]:** Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O mode.

00: Input mode (reset state)

01: General purpose output mode

10: Alternate function mode

11: Analog mode

GPIO port output type register (GPIOx_OTYPER)

This is a 16-bit register where each bit denotes the 'type' of a single pin in the register. This register sets the type of output pins to either push-pull or open drain. For example if pin PC7 is configured as an output pin, clearing bit 7 (or leaving its state at zero) of the OTYPER register associated with GPIOC (GPIOC_TYPER), will set the output type of the GPIO output pin PC7 to "Push-Pull".

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **OTy[1:0]:** Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O output type.

0: Output push-pull (reset state)

1: Output open-drain

GPIO output speed register (GPIOx_OSPEEDR)

This is a 32-bit register where each set of two bits represent the speed of a single output pin. For example bits 0 and 1 of the OSPEEDR register associated with port C (GPIOC_OSPEEDR), represent the speed setting of the output pin PC0 and bits 26 and 27 of the same register represent the speed setting of the output pin PC13. These two bits can be set to:

- 'x0': 2MHz Low speed
- '01':10MHz Medium speed
- '11': 50MHz High speed

So why have a speed setting on I/O ? To save power. On the 2MHz setting the GPIO would consume less current than on the 50MHz setting I'd imagine but would have relatively longer rise/fall time specs.

GPIO pull-up/pull-down register (GPIOx_PUPDR)

The GPIOx_PUPDR registers configures the internal pull-ups and pull-down resistors on each I/O pin. The internal pull-up/down resistors can be configured on GPIO pins set as input or output.The Pullup/down resistors have a typical value of 40 kOhms but can range from 30-50 kOhms.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]	
rw	rw	rw	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
PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 2y+1:2y **PUPDRy[1:0]:** Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O pull-up or pull-down

00: No pull-up, pull-down

01: Pull-up

10: Pull-down

11: Reserved

GPIO port input data register (GPIOx _IDR)

This is a 16-bit read-only register. Each bit represents the input value on a corresponding pin. Reading a '0' in bit 8 of this GPIOC _IDR register indicates that the voltage on PC8 is 0V (GND). While reading a '1' in bit 8 of this GPIOC _IDR register indicates that the voltage on PC8 is 3.3V (VDD).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **IDR[15:0]:** Port input data

These bits are read-only. They contain the input value of the corresponding I/O port.

GPIO port output data register (GPIOx_ODR)

This is a 16-bit read/write register. Each bit represents the output value on a corresponding pin. Writing a '0' in bit 8 of this GPIOC _ODR register indicates that the voltage on PC8 is driven by the micro to 0V (GND). While writing a '1' in bit 8 of this GPIOC _ODR register indicates that the voltage on PC8 is driven by the micro to 3.3V (VDD).

General Purpose I/O Port