Hello, and welcome to this presentation of the STM32G4 System Configuration Controller.
STM32G4 microcontrollers feature a set of configuration registers located in the SYSCFG module. The System Configuration Controller gives access to the following features:

- Remapping memory areas and selecting the memory accessible at address 0x0000_0000
- Managing the external interrupt line connection to the GPIOs
- Managing robustness feature
- Setting CCM RAM write protection and software erase
- Configuring FPU interrupts
- Enabling/disabling I2C Fast-mode Plus driving capability on some I/Os and voltage booster for I/Os analog switches.
This slide describes the memory map of the STM32G474 microcontroller.
The differences between the STM32G474 and STM32G431 variants are highlighted, later in this presentation.
The Flash memory size is up to 512 Kbytes, in a dual-bank configuration.
The FB_MODE bit determines the address mapping of Banks 1 and 2 and also selects which bank is aliased to address 0.
The SRAM total size is 128 Kbytes. It is split into 3 parts:
- The SRAM1 size is 80 Kbytes starting from address 0x2000_0000
- The SRAM2 size is 16 Kbytes starting from address 0x2001_8000
- The Core-Coupled Memory (or CCM) SRAM size is 32Kbytes starting from address 0x1000_0000.
SRAM1 and SRAM2 memories are located in the usual Arm V7-M memory space dedicated to SRAM while the CCM SRAM is accessed through Dcode and Icode AHB buses. This architecture enables concurrent accesses to CCM SRAM and SRAM1 or SRAM2 memories.
The memory remap at address 0 boosts up the code execution performance thanks to the dedicated ICODE and DCODE bus accesses, instead of using the System bus.

The memory remap at address 0 selects the memory accessible at address 0. It could be either the main Flash memory, or the system Flash memory, or the FMC bank 1, or the SRAM1, or the QUADSPI.

The FMC bank1 maps external NOR flash or PSRAM memory.

The FB_MODE bit in the System Configuration Remap register allows the swap in between Flash memory banks 1 and 2.
This figure represents the STM32G4 bus matrix. The bus masters are at the top of the figure: the Cortex-M4 core with its three AHB master interfaces ICODE, DCODE and SYSTEM and the two DMA controllers. The bus slaves are at the right of the figure: internal and external memories and peripherals. The Flash memory is read through the accelerator. When the Cortex-M4 core accesses data within the CODE or DATA address range, the DCODE bus is used. When the Cortex-M4 core accesses instructions within the CODE or DATA address range, the ICODE bus is used. The SRAM1 is accessed by default through the System bus, and can be accessed through the ICODE bus and DCODE bus when it is remapped at address 0, in order to increase performance. The CCM SRAM memory is always accessed through
the I-bus and D-bus allowing zero-wait-state code execution.
The Quad-SPI and FMC banks can be read and executed through the System bus by default, and can be remapped at 0 to increase performance.
The two DMA controllers can access all memories and peripherals.
This table compares the code execution performance at 150 MHz and 170 MHz while running the EEMBC CoreMark benchmark. The maximum performance is reached when the code is executed in CCM SRAM with data is located in SRAM1. It is also possible to reach maximum performance with code in SRAM1 and data in SRAM2 if the SRAM1 is remapped at address 0. When executing from Flash memory the maximum CoreMark performance is reached when the ART accelerator is enabled, and there is almost no loss of performance due to the Flash access time requiring 7 wait states at 150 MHz or 8 wait states at 170 MHz. Enabling the prefetch buffer yields a slightly higher score: 3.36 CoreMark / MHz in case of Single bank mode.
This slide presents the key differences between the STM32G431 and STM32G474 microcontrollers. The STM32G431 line includes neither QuadSPI nor FMC units.

The STM32G431 line has smaller SRAM memories: a 16-Kilobyte SRAM1 and a 10-kilobyte CCM SRAM both supporting parity + a 6-kilobyte SRAM2.

Regarding the mapping, the CCM SRAM is aliased at address 0x2000_5800 to allow continuous RAM address range with SRAM1 and SRAM2 memories.

At last, the STM32G431 has a unique bank of flash memory.
There are 3 boot modes which are selected by the BOOT0 pin or the nBOOT0 bit (if the nSWBOOT0 bit is cleared) and by nBOOT1 bit.

When the BOOT0 pin is at a low level, the STM32G4 microcontroller boots from the User Flash memory.

When the BOOT0 pin is at a high level, the nBOOT1 bit determines the boot mode.

When it is high, the boot is done from the system memory, that contains the ST proprietary boot code.

The other option is booting from the SRAM1 memory region.

When the boot lock option bit is high, the boot is forced from the Main Flash memory.

Software can dynamically select which memory is visible at address 0 by programming the MEM_MODE field in the SYSCFG_MEMRMP register.

The default value of this field depends on the boot pins.
state and related option bytes value, BOOT_LOCK and nSWBOOT0.
The on-chip boot-loader allows the user to program the Flash memory with an image downloaded to the STM32G4 through a serial communication peripheral. The supported protocols are USART, USB, CAN, SPI and I²C.
The 32 Kbytes of CCM SRAM is particularly suitable for performance, integrity and safety. The CCM SRAM is accessed through the DCODE and ICODE buses without any remapping, which enables code execution at zero-wait-states. The CCM SRAM supports parity check. The Data bus width is 36 bits because 4 bits are available for parity check (1 bit per byte) in order to increase the memory robustness, as required, for instance, by Class B or SIL standards. Class B and SIL are safety standards: Class B is for Home Appliances and SIL for the Safety Integrity Level.

The parity bits are computed and stored when writing into the SRAM. Then, they are automatically checked when reading. If at least one bit fails, a Non-Maskable Interrupt (NMI) is generated. The same error can also be linked to the Break input of the timers and the HRTIM.
system fault input. 
Note that the parity check is disabled by default. 
The lower 32 Kbytes of the SRAM1 also support parity 
generation and checking.
The CCM SRAM is also suitable for secure applications. It can be write-protected with a 1-Kbyte granularity. It can also be readout-protected via the RDP option byte. When protected, the CCM SRAM as well as the Flash main memory and the backup registers are totally inaccessible in debug mode or when code is running from boot RAM or boot loader.

The CCM SRAM is erased when the readout protection is changed from Level 1 to Level 0.

The CCM SRAM can be erased by software by setting the CCMER bit in the CCM SRAM System Configuration Control and Status register.

The CCM SRAM can also be erased with the system reset depending on the CCMSRAM_RST option bit in the user option bytes.
The System Configuration Register 2 contains the control and status bits related to safety and robustness such as the CCM SRAM parity error flag, and the control bits to direct some error detections events to the timers’ break inputs. This allows timer outputs to be placed in a known state during an application crash. Once programmed, the connection is locked until the next system reset. These internal events include a Flash error-code-correction event, a power voltage detector event, SRAM1 and CCM RAM parity error event, and the Cortex M4 hard fault.
The Floating Point Unit (FPU) present in the Cortex-M4 core sets the cumulative exception status flag in the FPSCR register as required for each instruction, in accordance with the FPv4 architecture. The FPU does not support user-mode traps. The exception enable bits in the FPSCR read-as-zero, and writes are ignored. The processor also has six output pins: IXC, FC, OFC, DZC, IDC, and IOC, that each reflects the status of one of the cumulative exception flags. When the corresponding enable bit in the SYSCFG_CFRGR1 register is set, an interrupt is requested when the flag is set. The interrupt service routine is in charge of determining which flag or flags have been set.
The four I2C controllers embedded in the STM32G4 microcontroller support 3 speeds:
- Standard-mode, the maximum bitrate is 100 Kilobits per second,
- Fast-mode, the maximum bitrate is 400 Kilobits per second,
- Fast-mode Plus, the maximum bitrate is 1 Megabit per second.

Fast-mode Plus requires a high drive capability, which is enabled in the SYSCFG module. Since high-drive is controlled at pin level, it is also available for the other alternate functions.
Each I2C controller has a control bit in the SYSCFG_CFGR1 register to enable fast mode plus driving capability mode. Each pin PB6, PB7, PB8, PB9 has its own I2C_PB_FMP control bit to activate the fast mode plus driving capability, whatever the selected alternate function. When FM+ mode is activated on the GPIO pin, the speed configuration of the GPIO, programmed in the GPIOx_OSPEEDR register, is ignored. In some cases, there are two ways to activate the FM+ mode. For instance, the PB8 pin is configured as I2C1_SCL supports FM+ when the I2C1_FMP or I2C_PB8_FMP bit is set to one in the SYSCFG_CFGR1 register.
Two bits from the SYSCFG_CFG1 register are used to select the power supply of the I/O analog switch: BOOSTEN and ANASWVDD. They have to be initialized according to the voltage of the VDD and VDDA power supplies.

- When VDDA voltage is larger than 2.4 V, the I/O analog switch should be powered by VDDA.
- When VDDA voltage is lower than 2.4 V and VDD is higher than 2.4 V, the I/O analog switch should be powered by VDD.
- When both VDD and VDDA voltages are lower than 2.4 V, the I/O analog switch should be powered by the output of the VDD booster.

<table>
<thead>
<tr>
<th>VDD</th>
<th>VDDA</th>
<th>BOOSTEN</th>
<th>ANASWVDD</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>&gt; 2.4 V</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>&gt; 2.4 V</td>
<td>&lt; 2.4 V</td>
<td>I/O analog switches are supplied by VDDA voltage</td>
<td>I/O analog switches supplied by VDDA</td>
</tr>
<tr>
<td>&lt; 2.4 V</td>
<td>&lt; 2.4 V</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I/O analog switches are supplied by a dedicated voltage booster (supplied by VDD)</td>
<td>I/O analog switches supplied by booster</td>
</tr>
</tbody>
</table>

![GPIO analog switch diagram](image-url)
The STM32G4 microcontroller has 7 IO ports: Ports A to F are 16-pin wide ports, port G is 11-pin wide port. Each of the 16 EXTI configurable events related to GPIO ports has an independent multiplexor. The EXTI multiplexer outputs are available independently from any masks defined in the EXTI_IMR and EXTI_EMR registers.

- Two or more GPIO pads having the same number in different ports cannot be selected at the same time as EXTI configurable events
For more details about the System Configuration module, refer to the reference manual for STM32G4 microcontrollers.
Refer also to these trainings for more information if needed:
- Extended interrupts and event Controller (EXTI)
- Arm Cortex-M4 core (CM4)
- Memory protection (MEMPROTECT)
- Timers (TIM)
- High Resolution Timer (HRTIM)