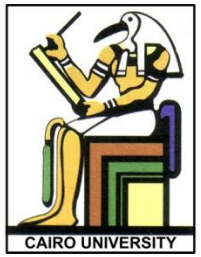


STM32 MICROCONTROLLER

Lecture 2

Prof. Yasser Mostafa Kadah

Harvard and von Neumann Architectures

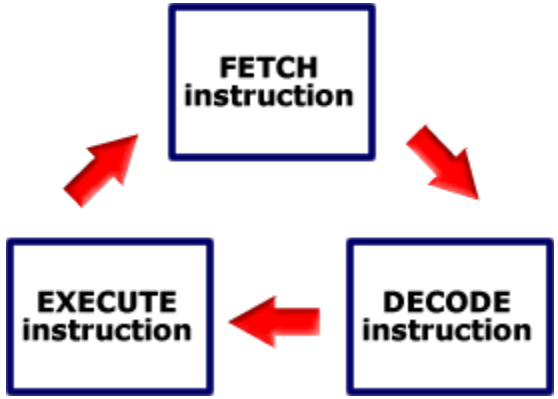
- Harvard Architecture—a type of computer architecture where the instructions (program code) and data are stored in separate memory spaces
  - Example: Intel 8051 architecture

- von Neumann Architecture—another type of computer architecture where the instructions and data are stored in the same memory space
  - Example: ARM, Intel x86 architecture (Intel Pentium, AMD Athlon, etc.)

Instruction Execution Cycle

- Fetch operation—retrieves an instruction from the location in code memory pointed to by the program counter (PC)

- Execute operation—executes the instruction that was fetched during the fetch operation. In addition to executing the instruction, the CPU also adds the appropriate number to the PC to point it to the next instruction to be fetched.



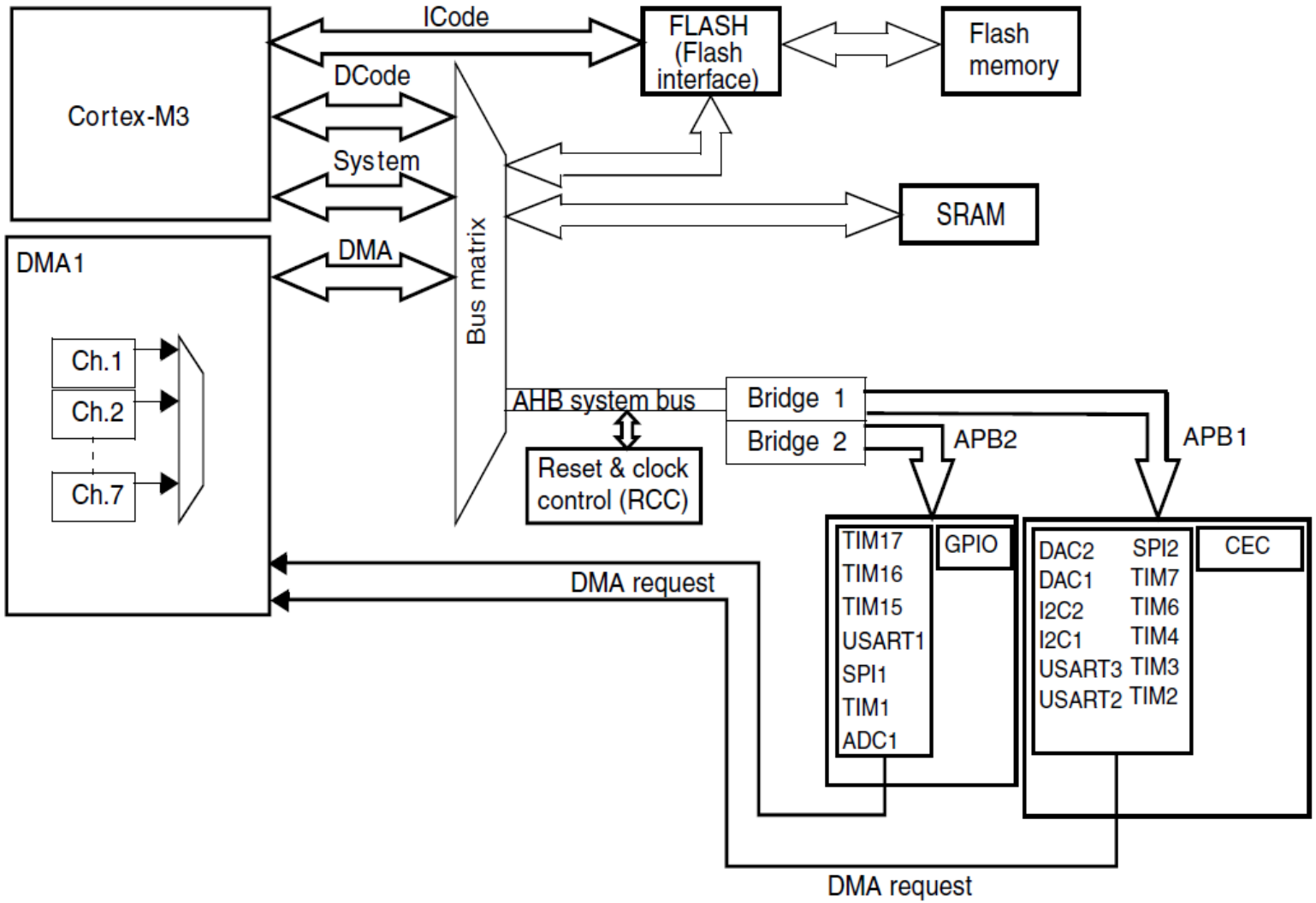
Microcontroller Architectures

- Microcontroller architecture refers to the internal hardware organization of a microcontroller
- Each hardware architecture has its own set of software instructions called assembly language that allows programming of the microcontroller
- Some of the popular microcontroller architectures
  - Intel 8051
  - Zilog Z80
  - Atmel AVR
  - Microchip PIC
  - ARM

Memory and Bus Architecture

- Three masters:
  - Cortex™-M3 core DCode bus (D-bus) and System bus (S-bus)
  - GP-DMA1 (general-purpose DMA)
- Three slaves:
  - Internal SRAM
  - Internal Flash memory
  - AHB to APB bridges (AHB to APBx), which connect all the APB peripherals

System Architecture



Definitions

- ICode bus
  - Connects the instruction bus of the Cortex™-M3 core to the Flash memory instruction interface. Instruction fetches are performed on this bus.
- DCode bus
  - Connects the DCode bus (literal load and debug access) of the Cortex™-M3 core to the Flash memory data interface
- System bus
  - Connects the system bus of the Cortex™-M3 core (peripherals bus) to a bus matrix which manages the arbitration between the core and the DMA

Definitions

- DMA bus
  - Connects the AHB master interface of the DMA to the bus matrix which manages the access of CPU DCode and DMA to the SRAM, Flash memory and peripherals
- Bus matrix
  - Manages the access arbitration between the core system bus and the DMA master bus. The arbitration uses a round robin algorithm
- AHB/APB bridges (APB)
  - The two AHB/APB bridges provide full synchronous connections between the AHB and the two APB buses
  - APB buses operate at full speed (up to 24 MHz)

Memory Organization

- Program memory, data memory, registers and I/O ports are organized within the same linear 4-Gbyte address space
- Bytes are coded in memory in *little endian* format
  - The lowest numbered byte in a word is considered the word's least significant byte and the highest numbered byte, the most significant
- Addressable memory space is divided into 8 main blocks, each of 512 MB

Memory Map

Boundary address	Peripheral	Bus
0x4002 3000 - 0x4002 33FF	CRC	AHB
0x4002 2400 - 0x4002 2FFF	Reserved	
0x4002 2000 - 0x4002 23FF	Flash memory interface	
0x4002 1400 - 0x4002 1FFF	Reserved	
0x4002 1000 - 0x4002 13FF	Reset and clock control RCC	
0x4002 0400 - 0x4002 0FFF	Reserved	
0x4002 0000 - 0x4002 03FF	DMA1	
0x4001 4C00 - 0x4001 FFFF	Reserved	APB2
0x4001 4800 - 0x4001 4BFF	TIM17 timer	
0x4001 4400 - 0x4001 47FF	TIM16 timer	
0x4001 4000 - 0x4001 43FF	TIM15 timer	
0x4001 3C00 - 0x4001 3FFF	Reserved	
0x4001 3800 - 0x4001 3BFF	USART1	
0x4001 3400 - 0x4001 37FF	Reserved	
0x4001 3000 - 0x4001 33FF	SPI1	
0x4001 2C00 - 0x4001 2FFF	TIM1 timer	
0x4001 2800 - 0x4001 2BFF	Reserved	
0x4001 2400 - 0x4001 27FF	ADC1	
0x4001 1C00 - 0x4001 23FF	Reserved	
0x4001 1800 - 0x4001 1BFF	GPIO Port E	
0x4001 1400 - 0x4001 17FF	GPIO Port D	
0x4001 1000 - 0x4001 13FF	GPIO Port C	
0x4001 0C00 - 0x4001 0FFF	GPIO Port B	
0x4001 0800 - 0x4001 0BFF	GPIO Port A	
0x4001 0400 - 0x4001 07FF	EXTI	
0x4001 0000 - 0x4001 03FF	AFIO	

Boundary address	Peripheral	Bus
0x4000 7C00 - 0x4000 FFFF	Reserved	APB1
0x4000 7800 - 0x4000 7BFF	CEC	
0x4000 7400 - 0x4000 77FF	DAC	
0x4000 7000 - 0x4000 73FF	Power control PWR	
0x4000 6C00 - 0x4000 6FFF	Backup registers (BKP)	
0x4000 5C00 - 0x4000 6BFF	Reserved	
0x4000 5800 - 0x4000 5BFF	I2C2	
0x4000 5400 - 0x4000 57FF	I2C1	
0x4000 4C00 - 0x4000 53FF	Reserved	
0x4000 4800 - 0x4000 4BFF	USART3	
0x4000 4400 - 0x4000 47FF	USART2	
0x4000 3C00 - 0x4000 3FFF	Reserved	
0x4000 3800 - 0x4000 3BFF	SPI2	
0x4000 3400 - 0x4000 37FF	Reserved	
0x4000 3000 - 0x4000 33FF	Independent watchdog (IWDG)	
0x4000 2C00 - 0x4000 2FFF	Window watchdog (WWDG)	
0x4000 2800 - 0x4000 2BFF	RTC	
0x4000 1800 - 0x4000 27FF	Reserved	
0x4000 1400 - 0x4000 17FF	TIM7 timer	
0x4000 1000 - 0x4000 13FF	TIM6 timer	
0x4000 0C00 - 0x4000 0FFF	Reserved	
0x4000 0800 - 0x4000 0BFF	TIM4 timer	
0x4000 0400 - 0x4000 07FF	TIM3 timer	
0x4000 0000 - 0x4000 03FF	TIM2 timer	

Memory Organization

- Embedded SRAM
  - The STM32F100xx features up to 32 Kbytes of static SRAM. It can be accessed as bytes, half-words (16 bits) or full words (32 bits). The SRAM start address is 0x2000 0000
- Bit banding
  - The Cortex™-M3 memory map includes two bit-band regions. These regions map each word in an alias region of memory to a bit in a bit-band region of memory. Writing to a word in the alias region has the same effect as a read-modify-write operation on the targeted bit in the bit-band region.
  - In the STM32F100xx, both peripheral registers and SRAM are mapped in a bit-band region. This allows single bit-band write and read operations to be performed.

Memory Organization

- Bit-Banding Mapping formula:

bit_word_addr = bit_band_base + (byte_offset x 32) + (bit_number × 4)

  - *bit_word_addr* is the address of the word in the alias memory region that maps to the targeted bit
  - *bit_band_base* is the starting address of the alias region
  - *byte_offset* is the number of the byte in the bit-band region that contains the targeted bit
  - *bit_number* is the bit position (0-7) of the targeted bit

Memory Organization

- Bit Banding Example
  - Mapping bit 2 of the byte located at SRAM address 0x2000 0300 in the alias region is done as follows:

    0x2200 6008 = 0x2200 0000 + (0x300*32) + (2*4).
  - Writing to address 0x2200 6008 has the same effect as a read-modify-write operation on bit 2 of the byte at SRAM address 0x2000 0300
  - Reading address 0x2200 6008 returns the value (0x01 or 0x00) of bit 2 of the byte at SRAM address 0x2000 0300 (0x01: bit set; 0x00: bit cleared)

Memory Organization

- Embedded Flash memory organization

Block	Name	Base addresses	Size (bytes)
Main memory	Page 0	0x0800 0000 - 0x0800 03FF	1 Kbyte
	Page 1	0x0800 0400 - 0x0800 07FF	1 Kbyte
	Page 2	0x0800 0800 - 0x0800 0BFF	1 Kbyte
	Page 3	0x0800 0C00 - 0x0800 0FFF	1 Kbyte
	Page 4	0x0800 1000 - 0x0800 13FF	1 Kbyte
	. . .	. . .	. . .
	Page 31	0x0800 7C00 - 0x0800 8000	1 Kbyte
Information block	System memory	0x1FFF F000 - 0x1FFF F7FF	2 Kbytes
	Option Bytes	0x1FFF F800 - 0x1FFF F80F	16
Flash memory interface registers	FLASH_ACR	0x4002 2000 - 0x4002 2003	4
	FLASH_KEYR	0x4002 2004 - 0x4002 2007	4
	FLASH_OPTKEYR	0x4002 2008 - 0x4002 200B	4
	FLASH_SR	0x4002 200C - 0x4002 200F	4
	FLASH_CR	0x4002 2010 - 0x4002 2013	4
	FLASH_AR	0x4002 2014 - 0x4002 2017	4
	Reserved	0x4002 2018 - 0x4002 201B	4
	FLASH_OBR	0x4002 201C - 0x4002 201F	4
	FLASH_WRPR	0x4002 2020 - 0x4002 2023	4

Boot Configuration

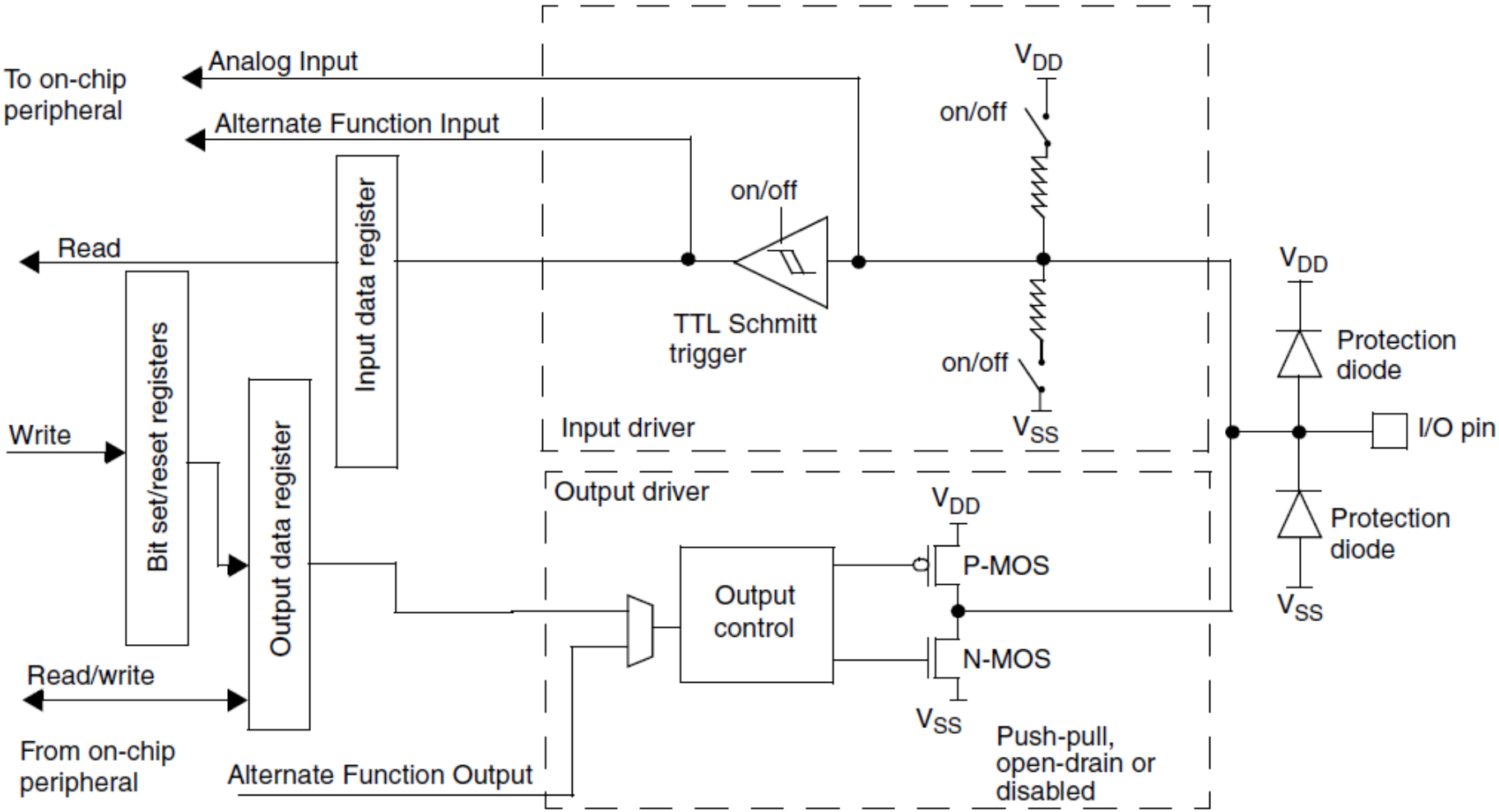
- Values on the BOOT pins are latched on the 4th rising edge of SYSCLK after a reset
  - It is up to the application to set the BOOT1 and BOOT0 pins after reset to select the required boot mode.
  - BOOT pins also resampled when exiting the Standby mode and hence must be kept in the required boot mode in the Standby mode
  - After this startup delay has elapsed, the CPU fetches the top-of-stack value from address 0x0000 0000, then starts code execution from the boot memory starting from 0x0000 0004.

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

General-Purpose I/Os (GPIOs)

- Each of the general-purpose I/O ports has:
  - Two 32-bit configuration registers (GPIOx_CRL, GPIOx_CRH)
  - Two 32-bit data registers (GPIOx_IDR, GPIOx_ODR)
  - 32-bit set/reset register (GPIOx_BSRR)
  - 16-bit reset register (GPIOx_BRR)
  - 32-bit locking register (GPIOx_LCKR)
- Each port bit of GPIOs can be individually configured by software in several modes:
  - Input floating
  - Input pull-up
  - Input-pull-down
  - Analog
  - Output open-drain
  - Output push-pull
  - Alternate function push-pull
  - Alternate function open-drain

Basic Structure of GPIO Bit



GPIO Bit Configuration Table

Configuration mode		CNF1	CNF0	MODE1	MODE0	PxODR register
General purpose output	Push-pull	0	0	01		0 or 1
	Open-drain		1	10		0 or 1
Alternate Function output	Push-pull	1	0	11		don't care
	Open-drain		1	See Table on following page		don't care
Input	Analog	0	0	00		don't care
	Input floating		1			don't care
	Input pull-down	1	0			0
	Input pull-up					1

- Note: During and just after reset, the alternate functions are not active and the I/O ports are configured in Input Floating mode (CNFx[1:0]=01b, MODEx[1:0]=00b)

GPIO Configuration: Output MODE Bits

MODE[1:0]	Meaning
00	Reserved
01	Max. output speed 10 MHz
10	Max. output speed 2 MHz
11	Max. output speed 50 MHz

GPIO Operation

- During and just after reset, the alternate functions are not active and the I/O ports are configured in Input Floating mode (CNFx[1:0]=01b, MODEx[1:0]=00b)
- When configured as output, the value written to the Output Data register (GPIOx_ODR) is output on the I/O pin.
  - It is possible to use the output driver in Push-Pull mode or Open-Drain mode (only the N-MOS is activated when outputting 0).
- The Input Data register (GPIOx_IDR) captures the data present on the I/O pin at every APB2 clock cycle
- All GPIO pins have an internal weak pull-up and weak pull-down which can be activated or not when configured as input

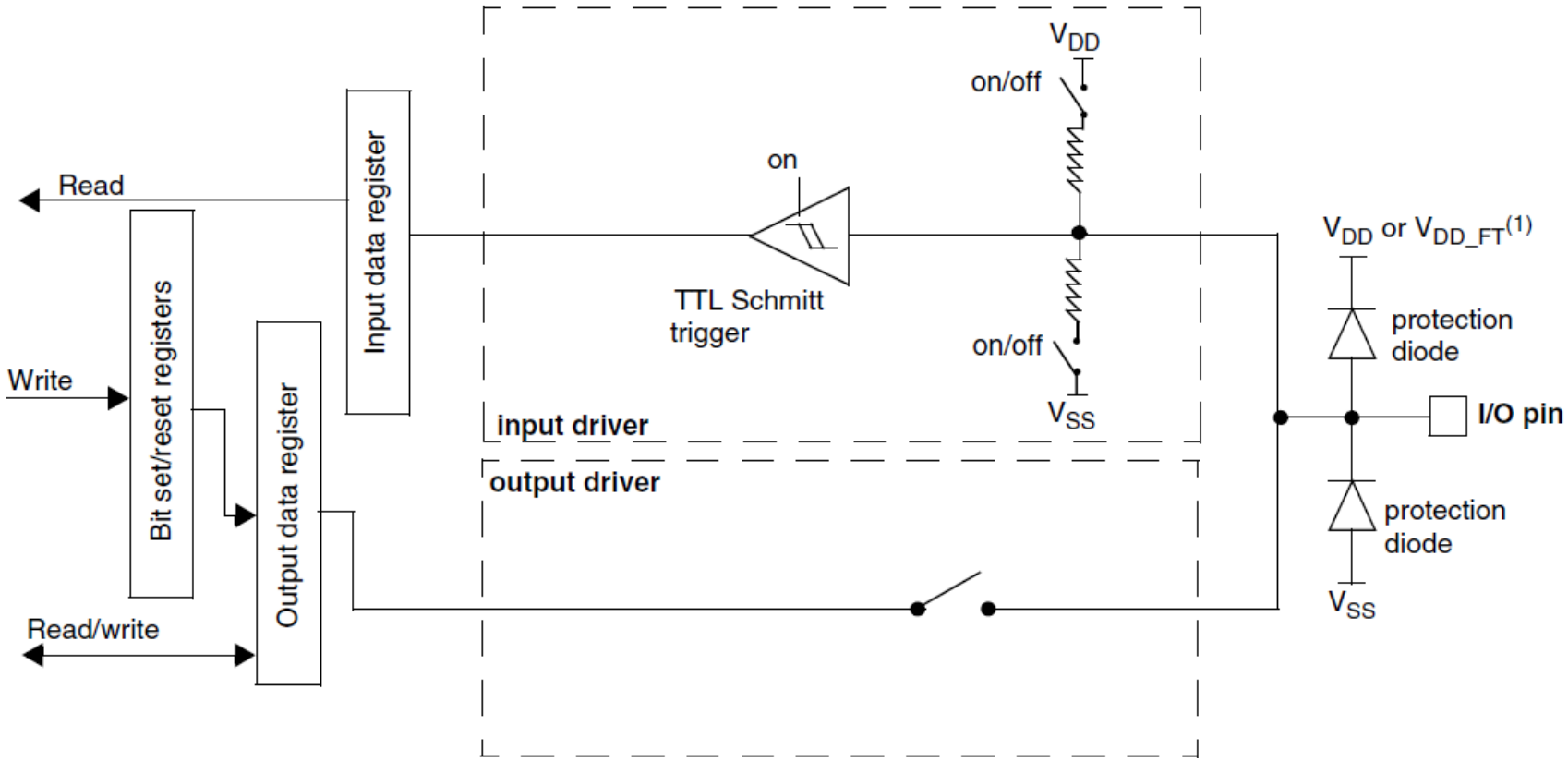
GPIO Atomic Bit Set or Reset

- Atomic Read/Modify access
  - No interruption in the middle to cause errors
- Atomic operations ensure that the desired change is not interrupted resulting in partial set/reset of GPIOs
- There is no need for the software to disable interrupts when programming the GPIOx_ODR at bit level: it is possible to modify only one or several bits in a single atomic APB2 write access
- This is achieved by programming to '1' the Bit Set/Reset Register (GPIOx_BSRR, or for reset only GPIOx_BRR) to select the bits you want to modify.
  - Unselected bits will not be modified

Input Configuration

- When the I/O Port is programmed as Input:
  - The Output Buffer is disabled
  - The Schmitt Trigger Input is activated
  - The weak pull-up and pull-down resistors are activated or not depending on input configuration (pull-up, pull-down or floating)
  - The data present on the I/O pin is sampled into the Input Data Register every APB2 clock cycle
  - A read access to the Input Data Register obtains the I/O State

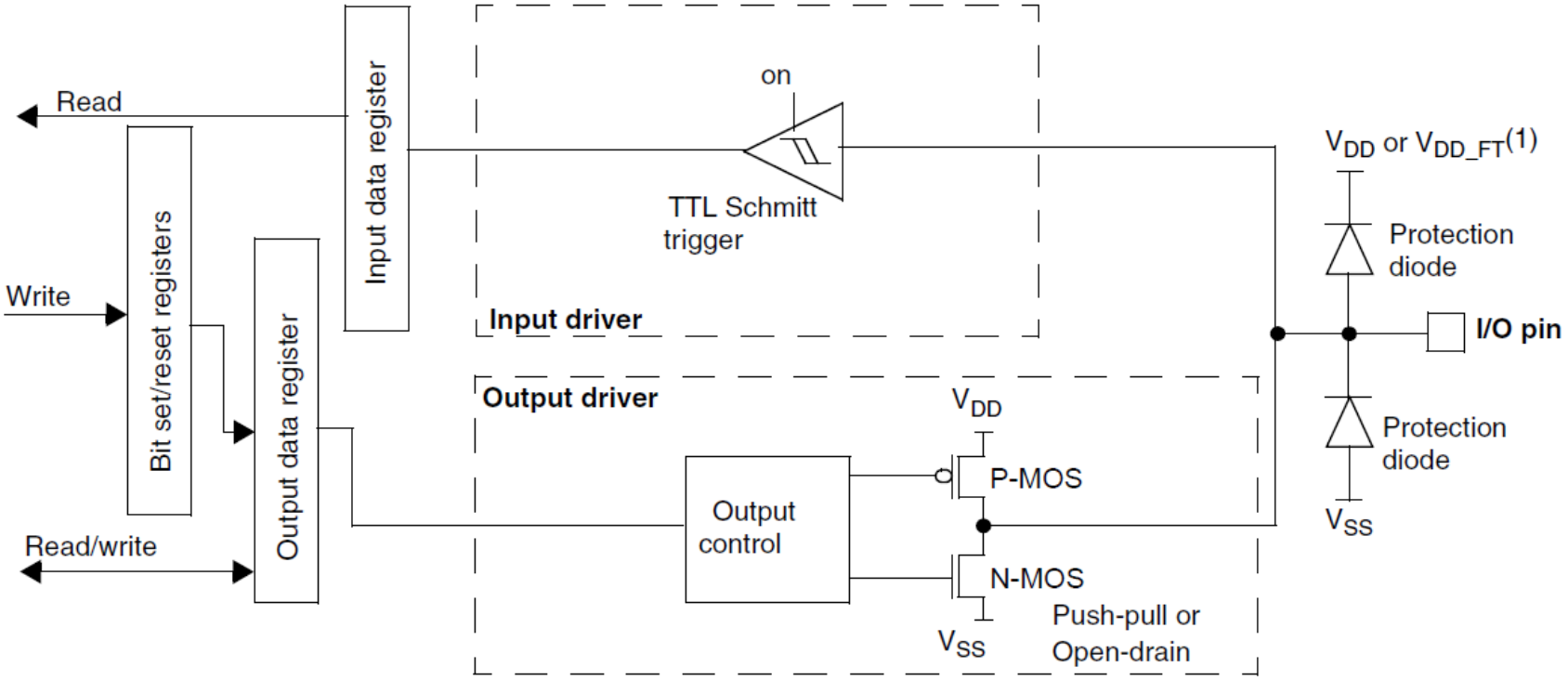
Input Configuration



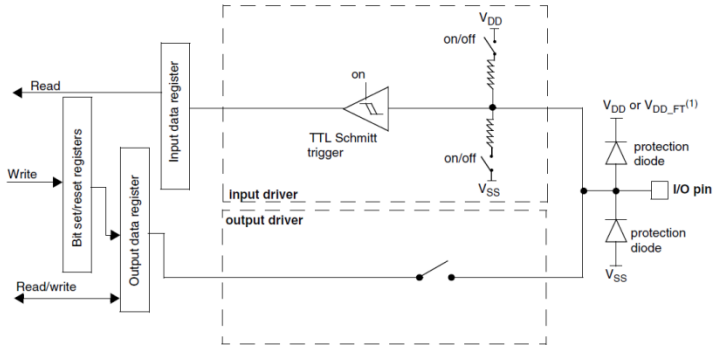
Output Configuration

- When the I/O Port is programmed as Output:
  - The Output Buffer is enabled:
    - Open Drain Mode: A “0” in the Output register activates the N-MOS while a “1” in the Output register leaves the port in Hi-Z. (the P-MOS is never activated)
    - Push-Pull Mode: A “0” in the Output register activates the N-MOS while a “1” in the Output register activates the P-MOS
  - The Schmitt Trigger Input is activated.
  - The weak pull-up and pull-down resistors are disabled.
  - The data present on the I/O pin is sampled into the Input Data Register every APB2 clock cycle
  - Read access to Input Data Register gets the I/O state in open drain mode
  - Read access to Output Data register gets last written value in Push-Pull mode

Output Configuration



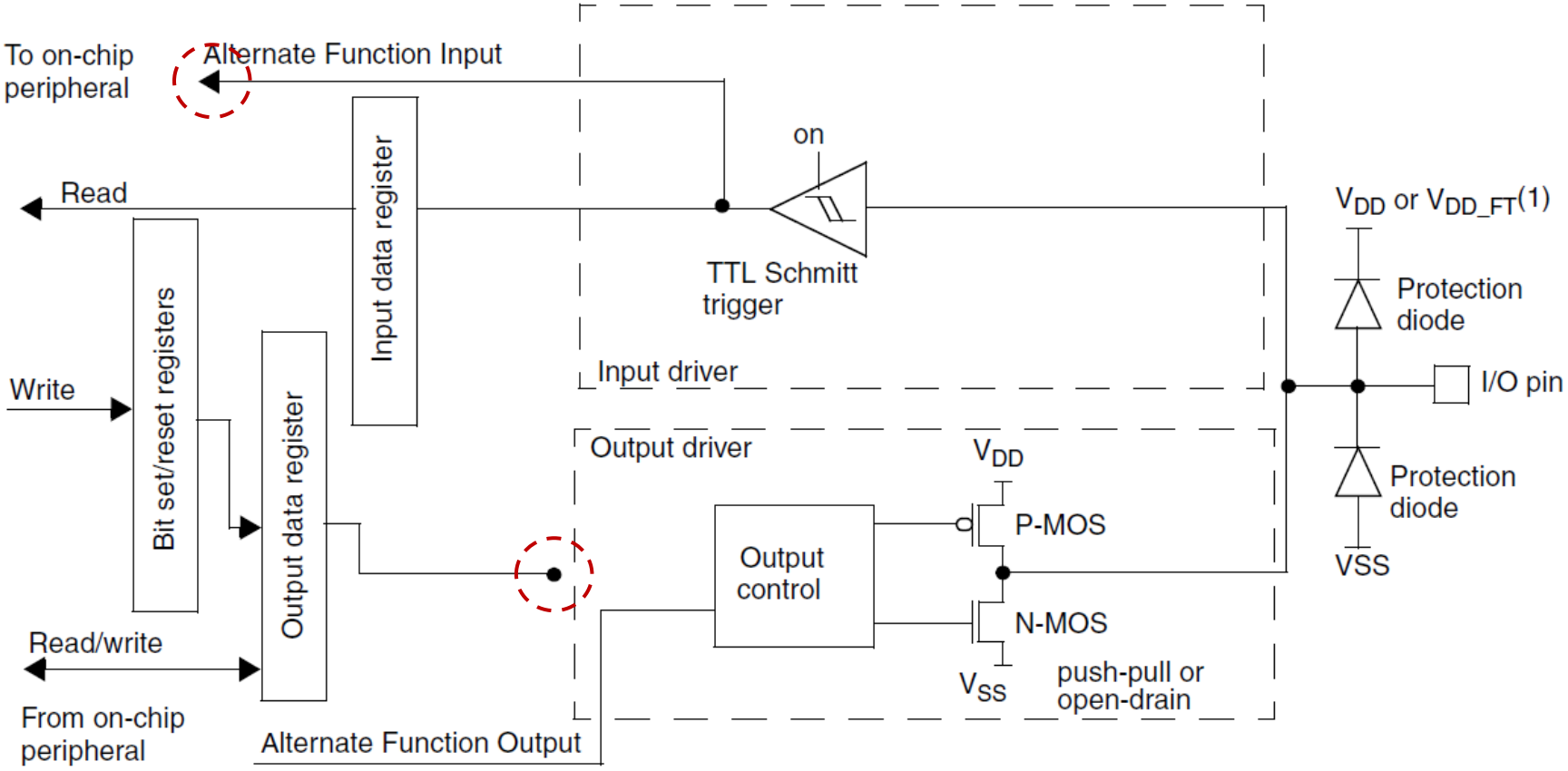
Compare to input configuration



Alternate Function Configuration

- When the I/O Port is programmed as Alternate Function:
  - The Output Buffer is turned on in Open Drain or Push-Pull configuration
  - The Output Buffer is driven by the signal coming from the peripheral (alternate function out)
  - The Schmitt Trigger Input is activated
  - The weak pull-up and pull-down resistors are disabled
  - The data present on the I/O pin is sampled into the Input Data Register every APB2 clock cycle
  - A read access to the Input Data Register gets the I/O state in open drain mode
  - A read access to the Output Data register gets the last written value in Push-Pull mode

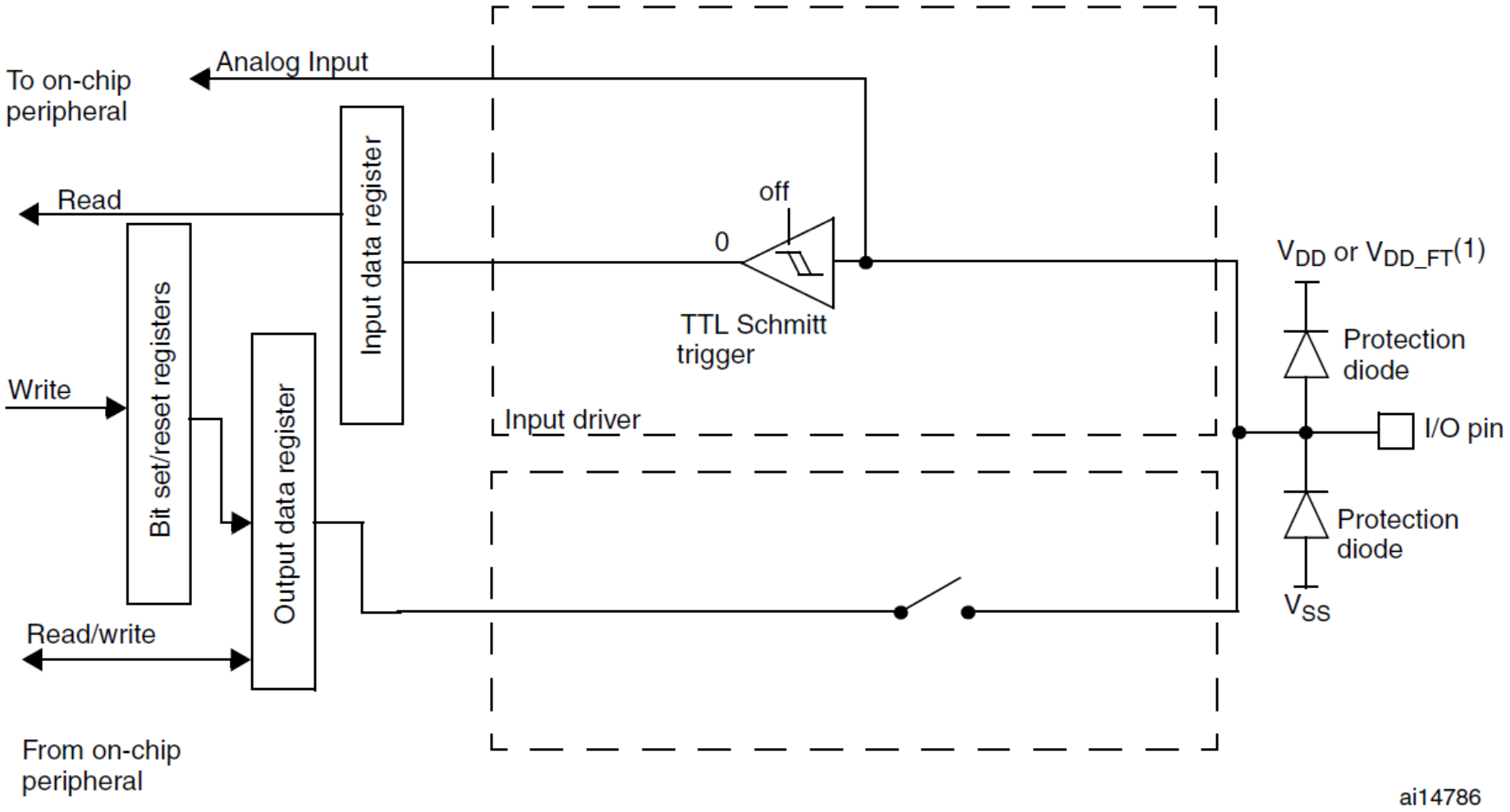
Alternate Function Configuration



Analog Configuration

- When the I/O Port is programmed as Analog configuration:
  - The Output Buffer is disabled.
  - The Schmitt Trigger Input is de-activated providing zero consumption for every analog value of the I/O pin. The output of the Schmitt Trigger is forced to a constant value (0).
  - The weak pull-up and pull-down resistors are disabled.
  - Read access to the Input Data Register gets the value “0”.

Analog Configuration



GPIO Registers

7.2.1 Port configuration register low (GPIOx_CRL) (x=A..G)

Address offset: 0x00

Reset value: 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7[1:0]		MODE7[1:0]		CNF6[1:0]		MODE6[1:0]		CNF5[1:0]		MODE5[1:0]		CNF4[1:0]		MODE4[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
CNF3[1:0]		MODE3[1:0]		CNF2[1:0]		MODE2[1:0]		CNF1[1:0]		MODE1[1:0]		CNF0[1:0]		MODE0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

7.2.2 Port configuration register high (GPIOx_CRH) (x=A..G)

Address offset: 0x04

Reset value: 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF15[1:0]		MODE15[1:0]		CNF14[1:0]		MODE14[1:0]		CNF13[1:0]		MODE13[1:0]		CNF12[1:0]		MODE12[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
CNF11[1:0]		MODE11[1:0]		CNF10[1:0]		MODE10[1:0]		CNF9[1:0]		MODE9[1:0]		CNF8[1:0]		MODE8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

GPIO Registers

7.2.3 Port input data register (GPIOx_IDR) (x=A..G)

Address offset: 0x08h

Reset value: 0x0000 XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
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

7.2.4 Port output data register (GPIOx_ODR) (x=A..G)

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

GPIO Registers

7.2.5 Port bit set/reset register (GPIOx_BSRR) (x=A..G)

Address offset: 0x10

Reset value: 0x0000 0000

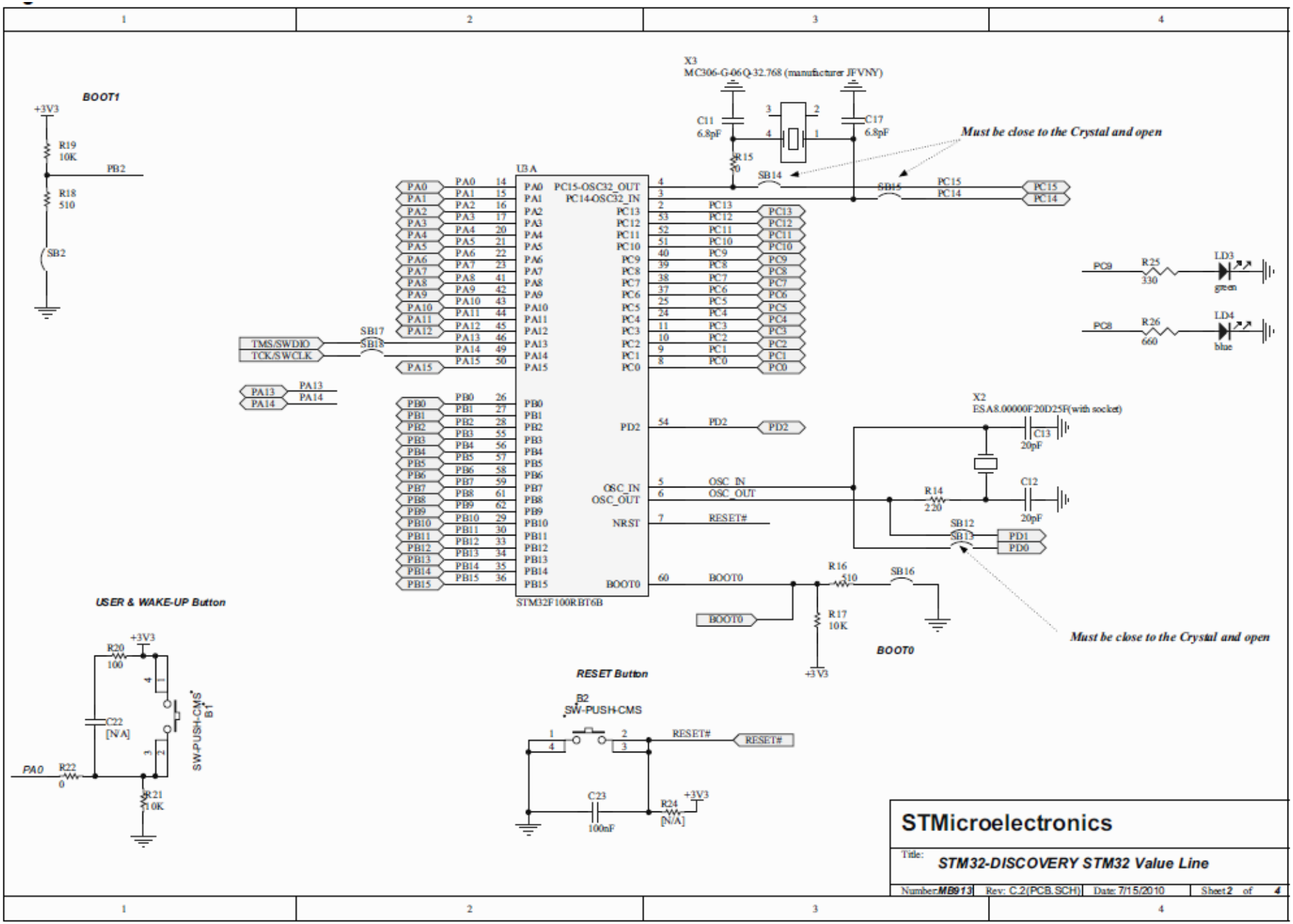
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

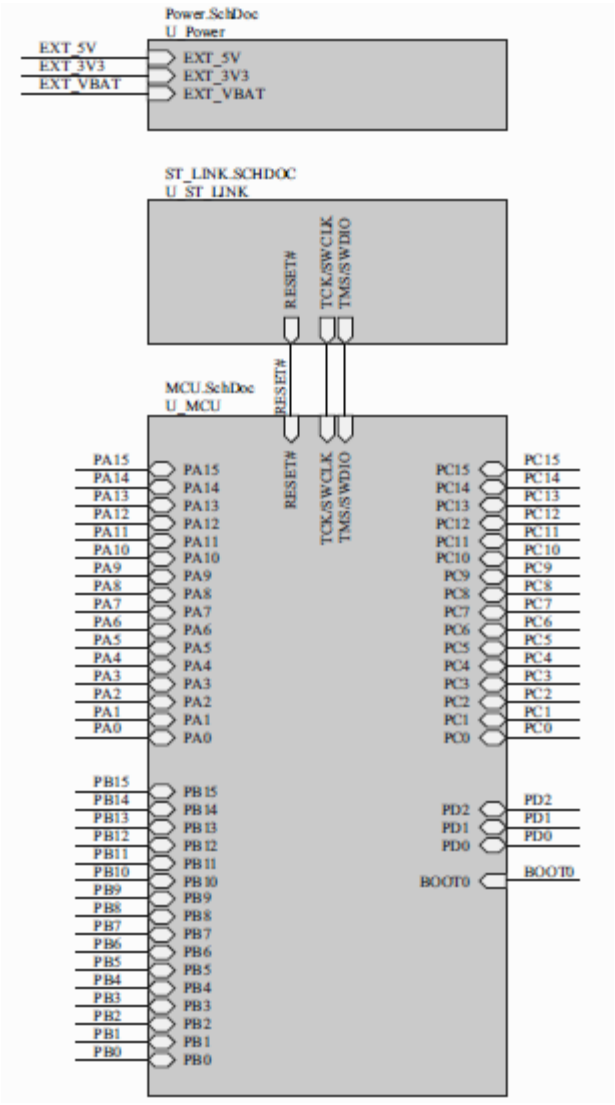
7.2.6 Port bit reset register (GPIOx_BRR) (x=A..G)

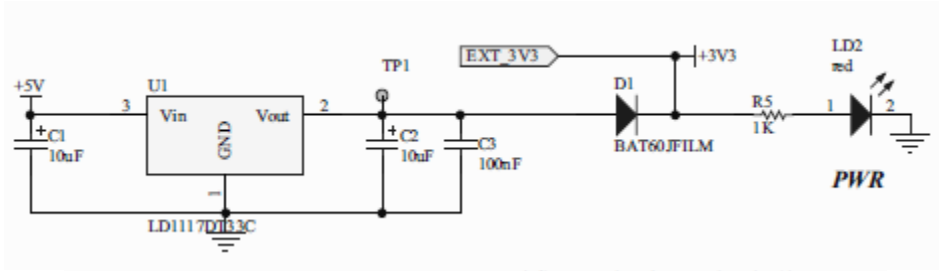
Address offset: 0x14

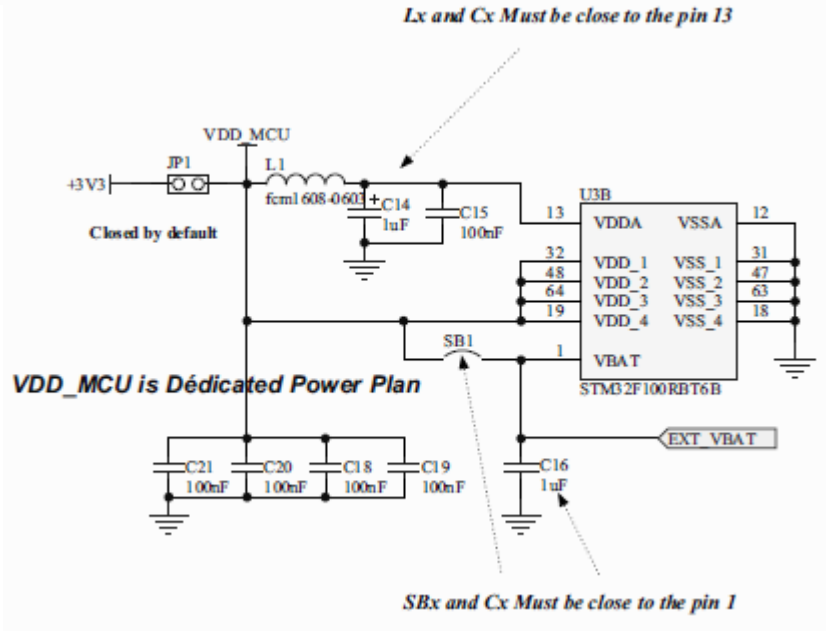
Reset value: 0x0000 0000

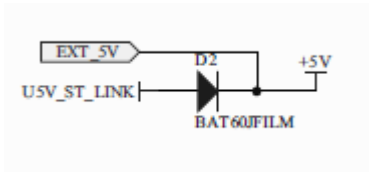
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

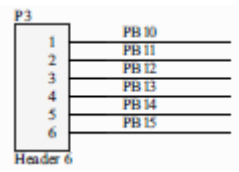

BOOT1
+3V3
R19
10K
PB2
R18
510
SB2
X3
MC306-G-06Q-32.768 (manufacturer JFVNY)
C11
6.8pF
C17
6.8pF
R15
0
SB14
Must be close to the Crystal and open
U3A
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PA11
PA12
PA13
PA14
PA15
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PB8
PB9
PB10
PB11
PB12
PB13
PB14
PB15
PC15-OSC32_OUT
PC14-OSC32_IN
PC13
PC12
PC11
PC10
PC9
PC8
PC7
PC6
PC5
PC4
PC3
PC2
PC1
PC0
PD2
OSC_IN
OSC_OUT
NRST
BOOT0
STM32F100RBT6B
SB15
SB17
SB18
TMS/SWDIO
TCK/SWCLK
LD3
green
LD4
blue
R25
330
R26
660
X2
ESA8.00000F20D25F(with socket)
C13
20pF
C12
20pF
R14
220
SB12
SB13
R16
510
SB16
R17
10K
USER & WAKE-UP Button
R20
100
C22
[N/A]
R22
0
R21
10K
SW-PUSH-CMS
B1
RESET Button
B2
SW-PUSH-CMS
C23
100nF
R24
[N/A]
STMicroelectronics
Title:
STM32-DISCOVERY STM32 Value Line
Number:MB913
Rev: C.2(PCB.SCH)
Date: 7/15/2010
Sheet 2 of 4

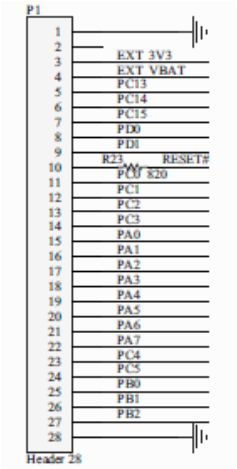

Power.SchDoc
U_Power
EXT_5V
EXT_3V3
EXT_VBAT
ST_LINK.SCHDOC
U_ST_LINK
RESET#
TCK/SWCLK
TMS/SWDIO
MCU.SchDoc
U_MCU
PA15
PA14
PA13
PA12
PA11
PA10
PA9
PA8
PA7
PA6
PA5
PA4
PA3
PA2
PA1
PA0
PB15
PB14
PB13
PB12
PB11
PB10
PB9
PB8
PB7
PB6
PB5
PB4
PB3
PB2
PB1
PB0
PC15
PC14
PC13
PC12
PC11
PC10
PC9
PC8
PC7
PC6
PC5
PC4
PC3
PC2
PC1
PC0
PD2
PD1
PD0
BOOT0

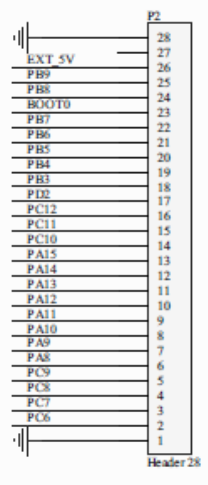

+5V
U1
Vin
GND
Vout
TP1
EXT_3V3
+3V3
D1
LD2
red
R5
1K
C1
10uF
C2
10uF
C3
100nF
BAT60JFILM
PWR
LD1117D1833C


Lx and Cx Must be close to the pin 13
VDD_MCU
JP1
+3V3
L1
fcm1608-0603
C14
1uF
C15
100nF
Closed by default
U3B
VDDA
VSSA
VDD_1
VDD_2
VDD_3
VDD_4
VSS_1
VSS_2
VSS_3
VSS_4
VBAT
SB1
VDD_MCU is Dédicated Power Plan
STM32F100RBT6B
EXT_VBAT
C21
100nF
C20
100nF
C18
100nF
C19
100nF
C16
1uF
SBx and Cx Must be close to the pin 1


EXT_5V
D2
+5V
U5V_ST_LINK
BAT60JFILM


P3
PB10
PB11
PB12
PB13
PB14
PB15
Header 6


P1
EXT_3V3
EXT_VBAT
PC13
PC14
PC15
PD0
PD1
R23
820
RESET#
PC0
PC1
PC2
PC3
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PC4
PC5
PB0
PB1
PB2
Header 28


P2
EXT_5V
PB9
PB8
BOOT0
PB7
PB6
PB5
PB4
PB3
PD2
PC12
PC11
PC10
PA15
PA14
PA13
PA12
PA11
PA10
PA9
PA8
PC9
PC8
PC7
PC6
Header 28

Assignments

- ARM Project #1 (To start this week)