Osnove mikroprocesorske elektronike

Marko Jankovec

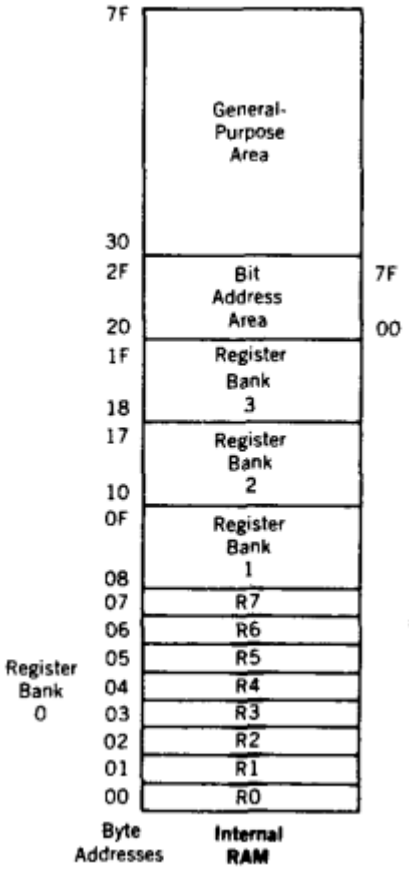
Arhitekture mikroprocesorjev

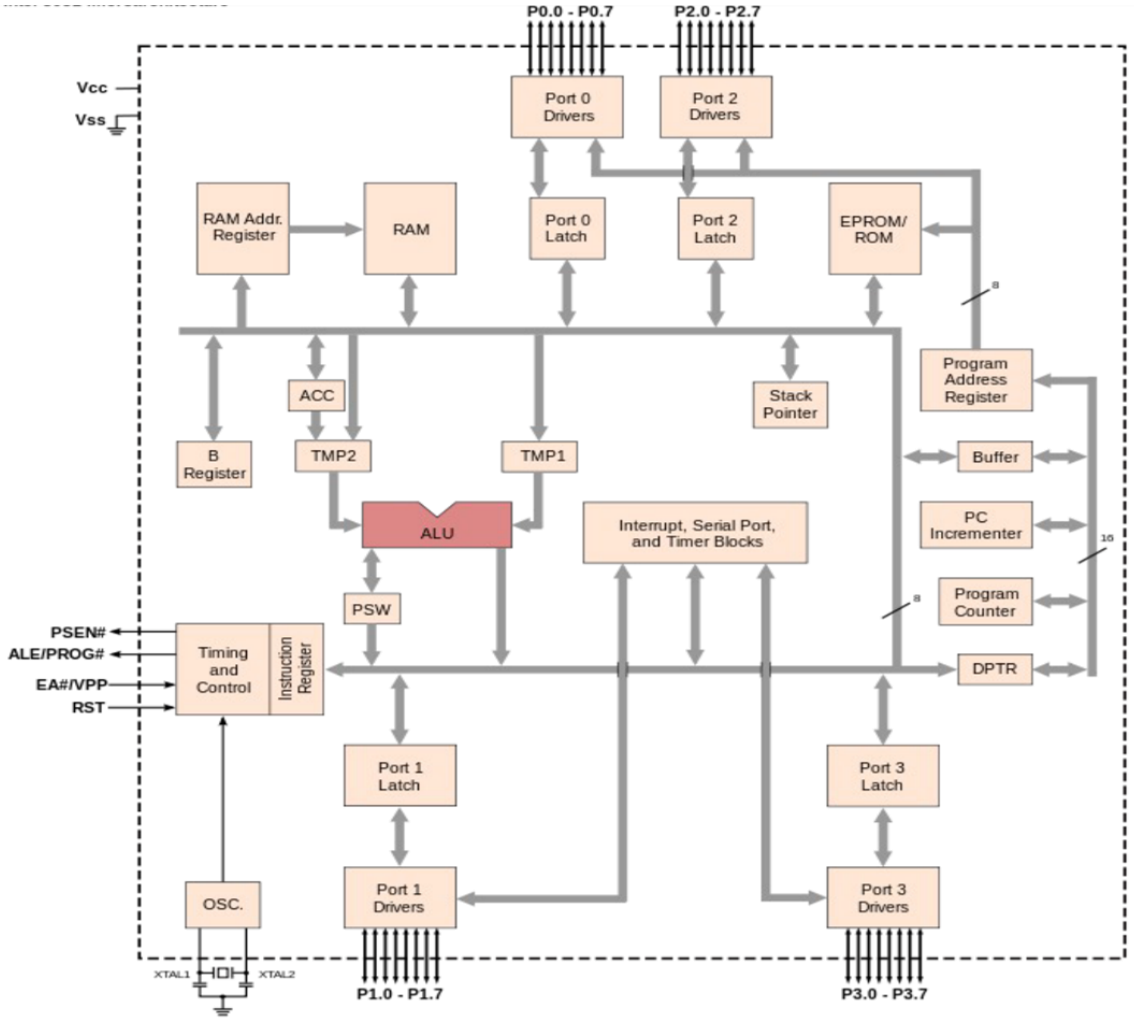
Primeri

Arhitekture najpogostejših mikrokrmilnikov

Družina	Arhitekura				Leto uvedbe
Intel 8051	Harvard	8-bit	RISC	akumulator	1980
Microchip PIC	Harvard	8-bit	RISC	akumulator	1985
Atmel AVR	Harvard	8-bit	RISC	registri	1996
Texas Instruments MSP430	Princeton	16-bit	RISC	registri	1992
ARM	Princeton /Harvard	32-bit	RISC	registri	1985

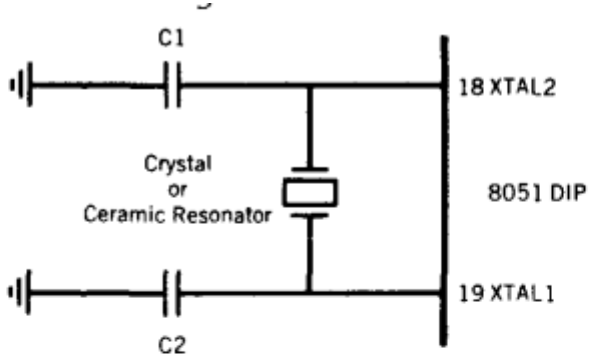
Intel 8051



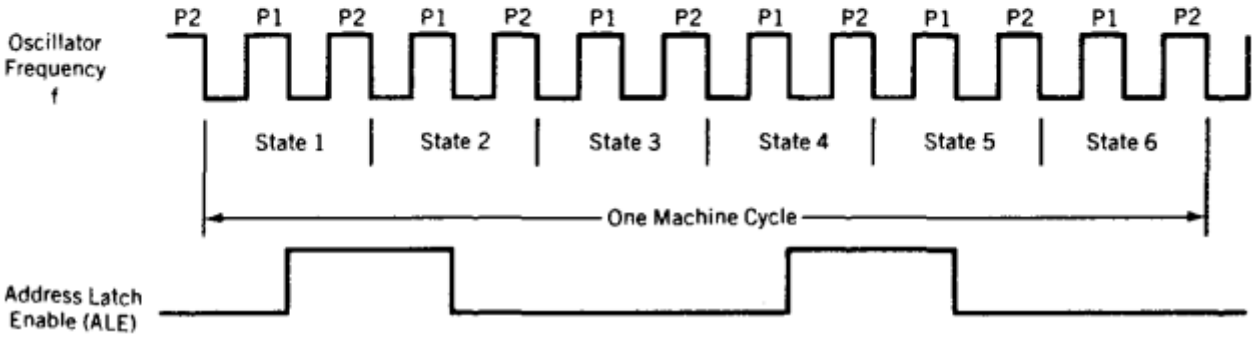


https://www.brainkart.com/article/Architecture-of-8051-Micro-Controller_7830/

Izvajanje inštrukcij

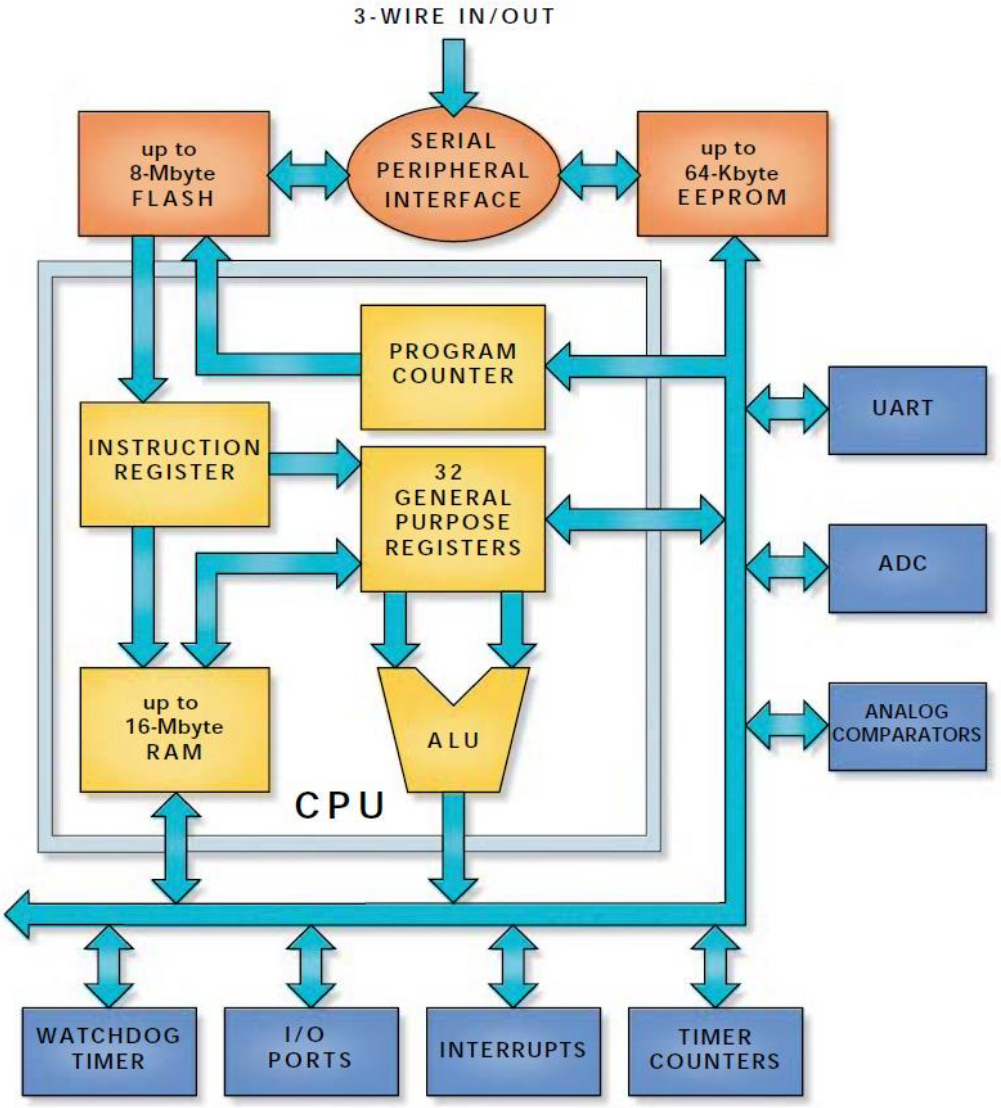


Crystal or Ceramic Resonator Oscillator Circuit



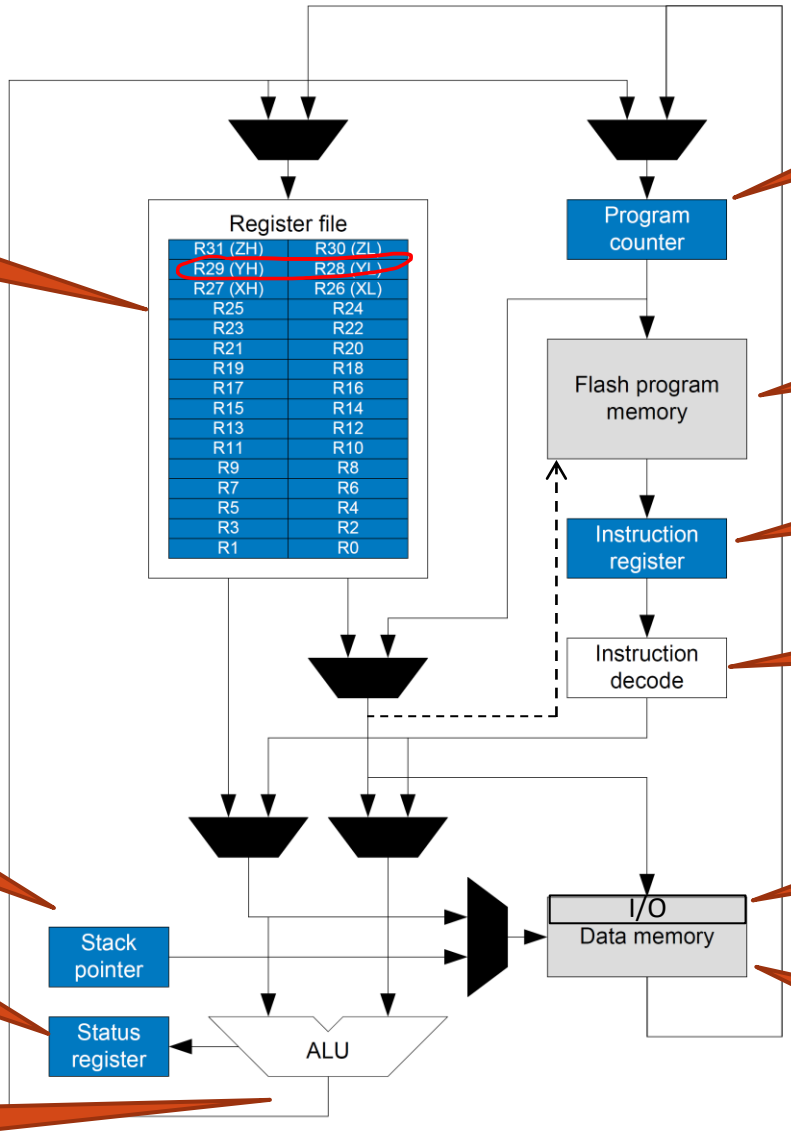
8051 Timing

Atmel AVR



http://lpvo.fe.uni-lj.si/fileadmin/files/Izobrazevanje/OME/microschematics/zarch.html

Jedro AVR



32 splošno namenskih registrov

Kaže na trenutno izvajano inštrukcijo

Pomnilnik tipa FLASH za programsko kodo

Hrani trenutno izvajano inštrukcijo

Usklajuje izvajanje inštrukcije

Kaže na prazno mesto v skladu

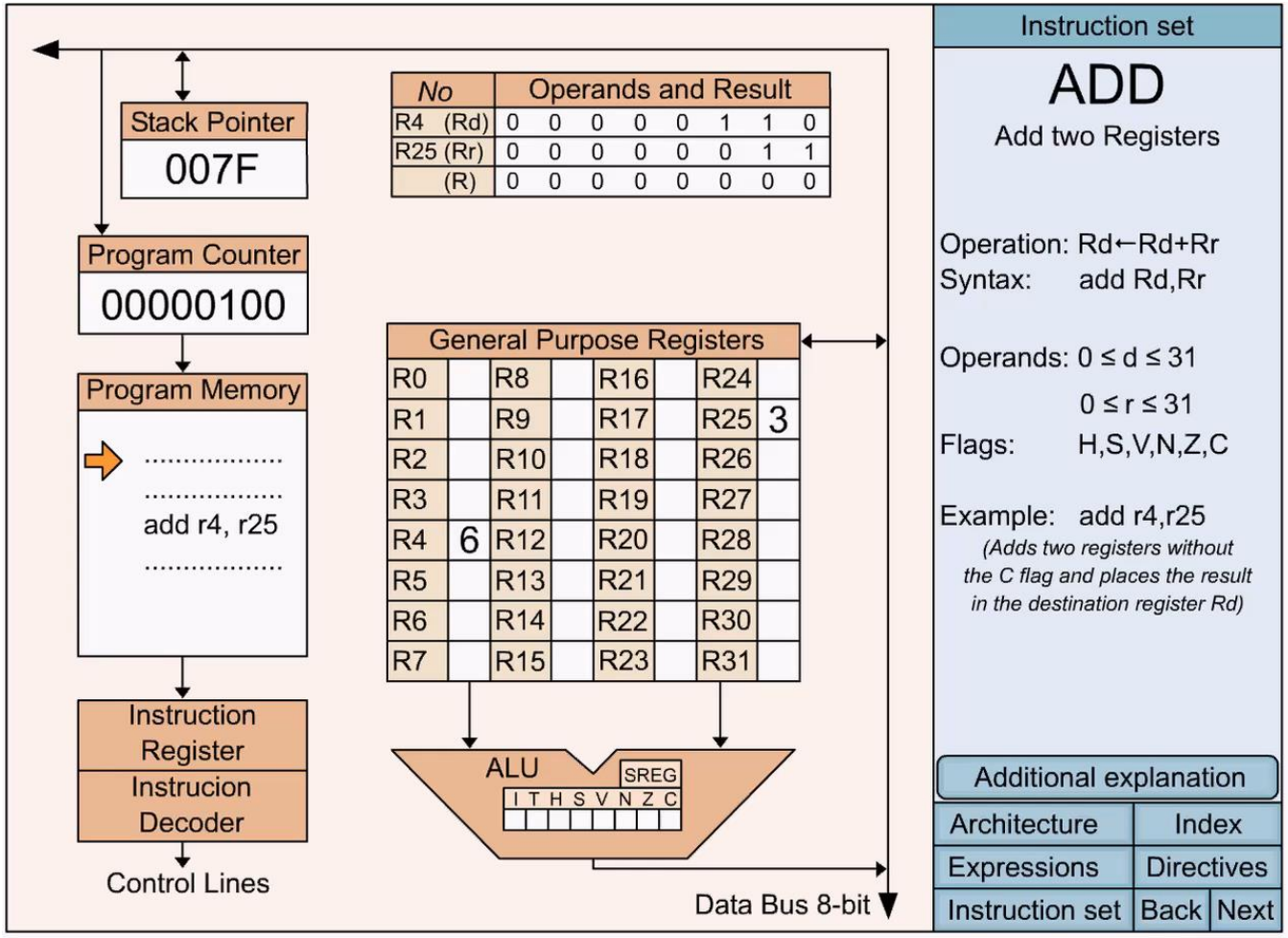
Vsebuje informacijo o izvedeni inštrukciji

Izvede inštrukcijo

I/O registri, ki krmilijo periferne I/O enote

Delovni pomnilnik tipa SRAM, ki hrani spremenljivke

Aritmetično-logična enota



Instruction set

ADD

Add two Registers

Operation: Rd←Rd+Rr
Syntax: add Rd,Rr

Operands: 0 ≤ d ≤ 31
0 ≤ r ≤ 31

Flags: H,S,V,N,Z,C

Example: add r4,r25
(Adds two registers without the C flag and places the result in the destination register Rd)

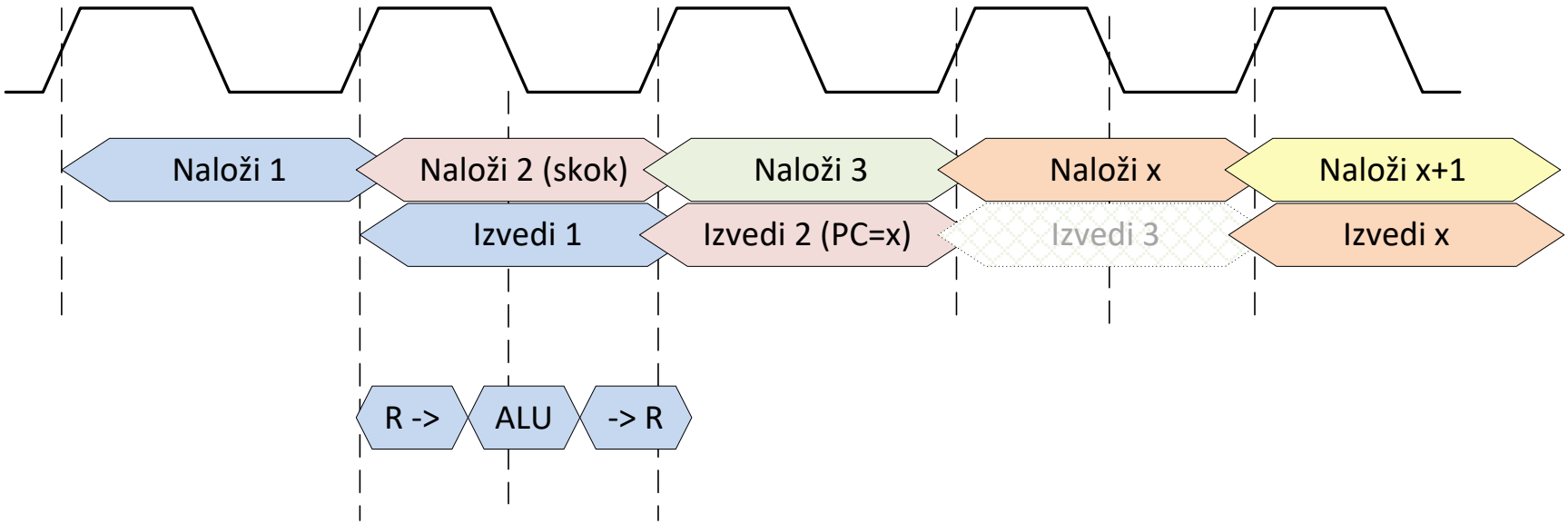
Additional explanation

Architecture	Index
Expressions	Directives
Instruction set	Back / Next

ADD	Add Two Registers
ADC	Add with Carry Two Registers
ADIW	Add Immediate to Word
LSL	Logical Shift Left
ROL	Rotate Left through Carry
AND	Logical AND Registers
ANDI	Logical AND Register and Constant
OR	Logical OR Registers
ORI	Logical OR Register and Constant
EOR	Exclusive OR Registers
COM	One's Complement
NEG	Two's Complement
SWAP	Swap Nibbles
INC	Increment
DEC	Decrement
SER	Set Register
CLR	Clear Register
SUB	Subtract Two Registers
SBC	Subtract with Carry Two Registers
SBIW	Subtract Immediate from Word
SUBI	Subtract Constant from Register
SBCI	Subtract with Carry Constant from Register
LSR	Logical Shift Right
ROR	Rotate Right through Carry
ASR	Arithmetic Shift Right
CP	Compare
CPC	Compare with Carry
CPI	Compare Register with Immediate
TST	Test for Zero or Minus

http://lpvo.fe.uni-lj.si/fileadmin/files/Izobrazevanje/OME/microschematics/add.html

Izvajanje inštrukcij - dvonivojsko cevovodenje

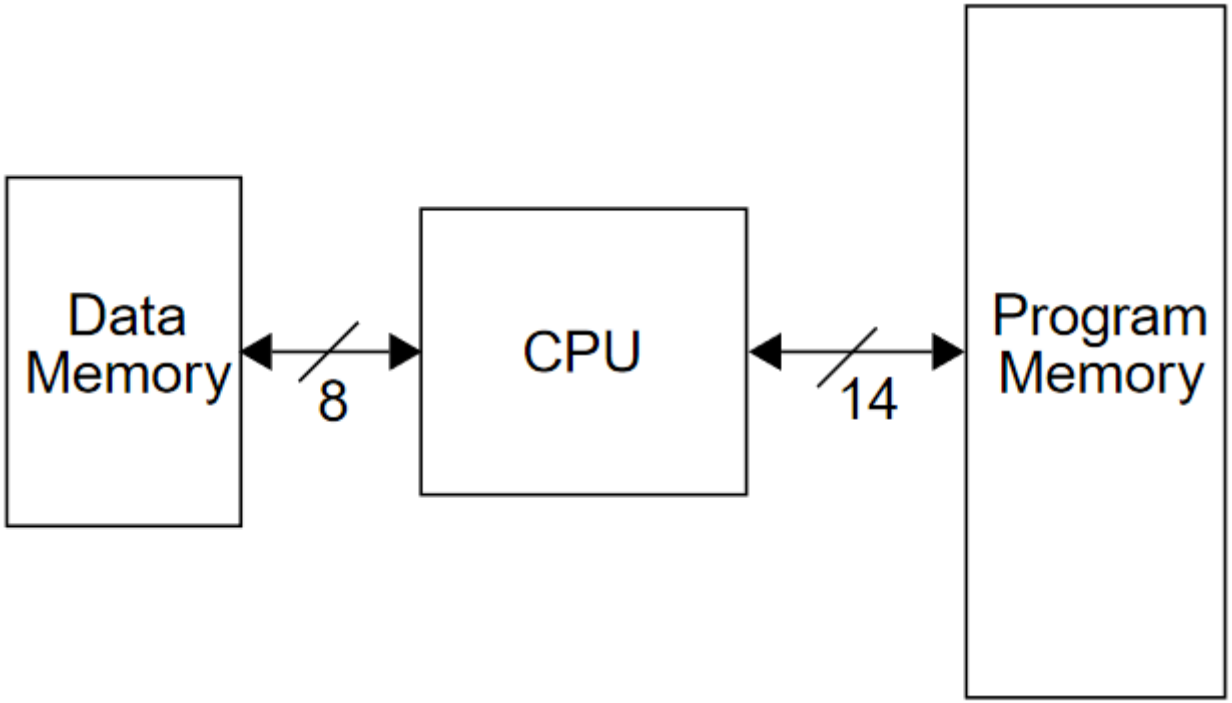


Programski pomnilnik (FLASH)

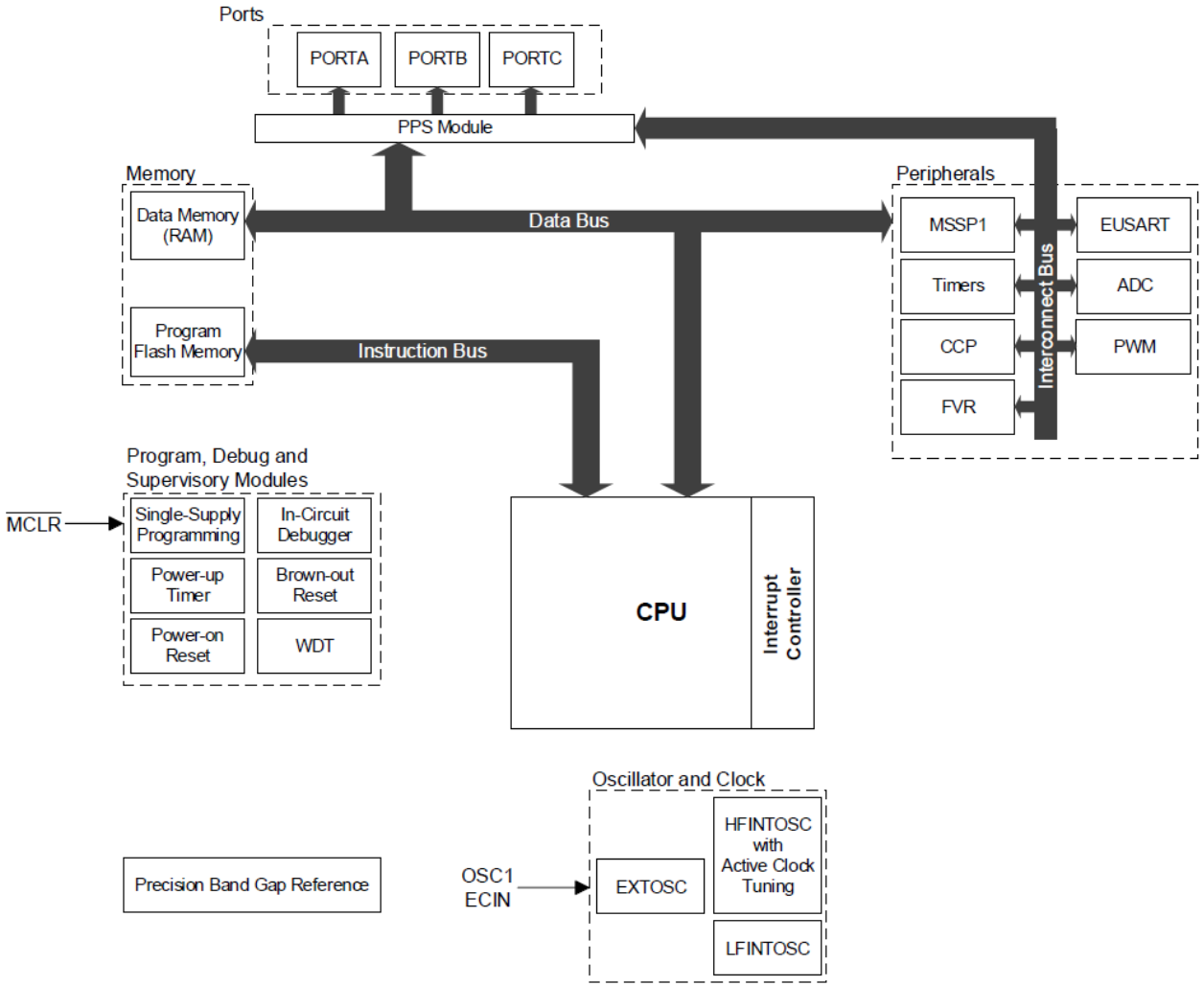
Word	Program Memory			Operation Code											
0000		ldi	r16 , 7	1	1	1	0	K	K	K	K	D	D	D	D	K	K	K	K
0001	loop:	inc	r17	1	0	0	1	0	1	0	D	D	D	D	D	0	0	1	1
0002		add	r16 , r17	0	0	0	0	1	1	R	D	D	D	D	D	R	R	R	R
0003		sbi	PORTB,6	1	0	0	1	1	0	1	0	A	A	A	A	A	B	B	B
0004		brbs	4, load	1	1	1	1	0	0	K	K	K	K	K	K	K	S	S	S
0005		in	r4,PORTB	1	0	1	1	0	A	A	D	D	D	D	D	A	A	A	A
0006		out	PORTB,r4	1	0	1	1	1	A	A	R	R	R	R	R	A	A	A	A
0007		sbrs	r16 , 0	1	1	1	1	1	1	1	R	R	R	R	R	0	B	B	B
0008		rjmp	loop	1	1	0	0	K	K	K	K	K	K	K	K	K	K	K	K
0009		ldd	r4, Y+2	1	0	Q	0	Q	Q	0	D	D	D	D	D	1	Q	Q	Q
000A	load	lds	r4, 27	1	0	0	1	0	0	0	D	D	D	D	D	0	0	0	0
000B				K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K
000C		st	Y, r4	1	0	0	0	0	0	1	R	R	R	R	R	1	0	0	0

http://lpvo.fe.uni-lj.si/fileadmin/files/Izobrazevanje/OME/microschematics/zpmemory.html

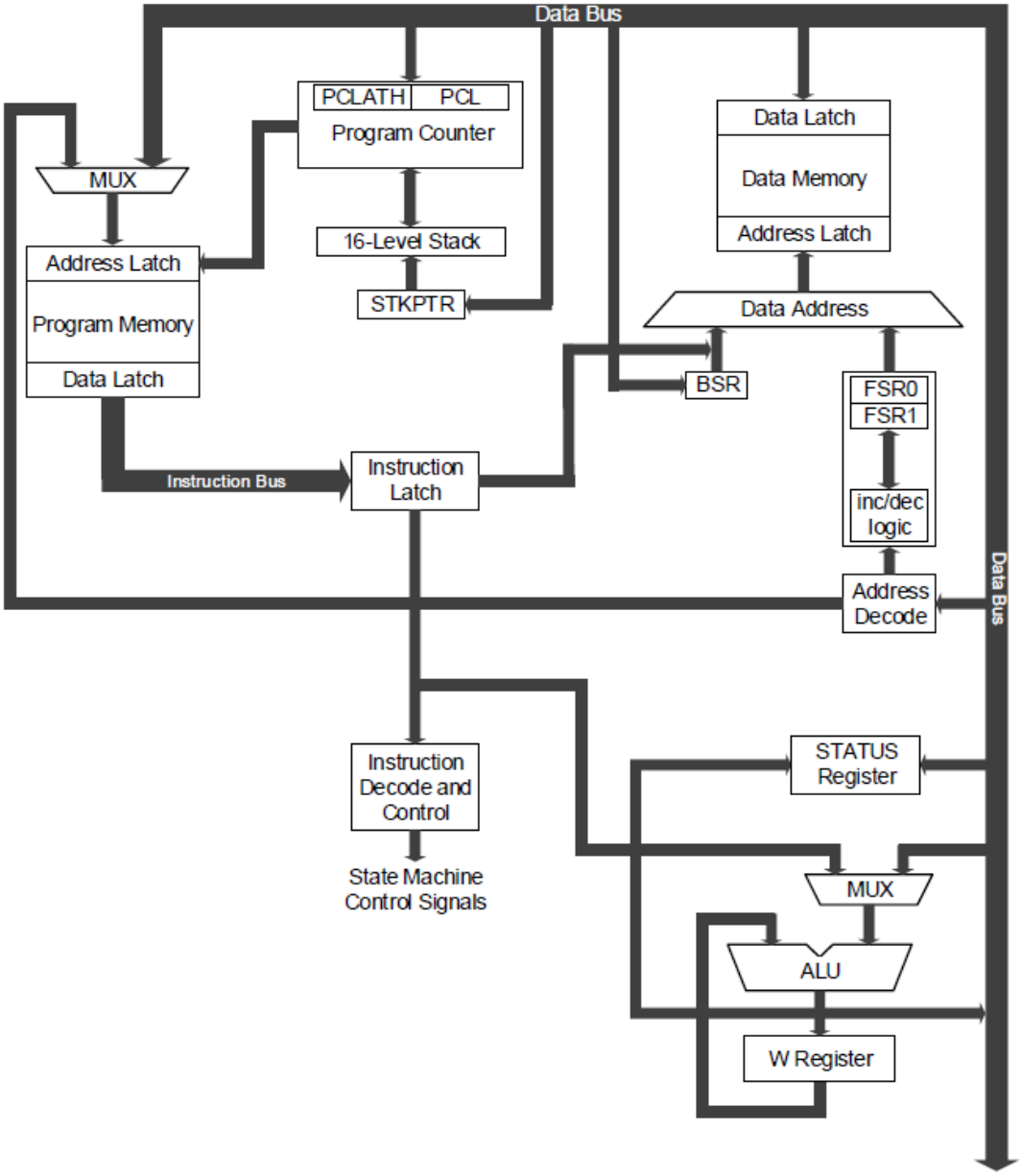
Microchip PIC



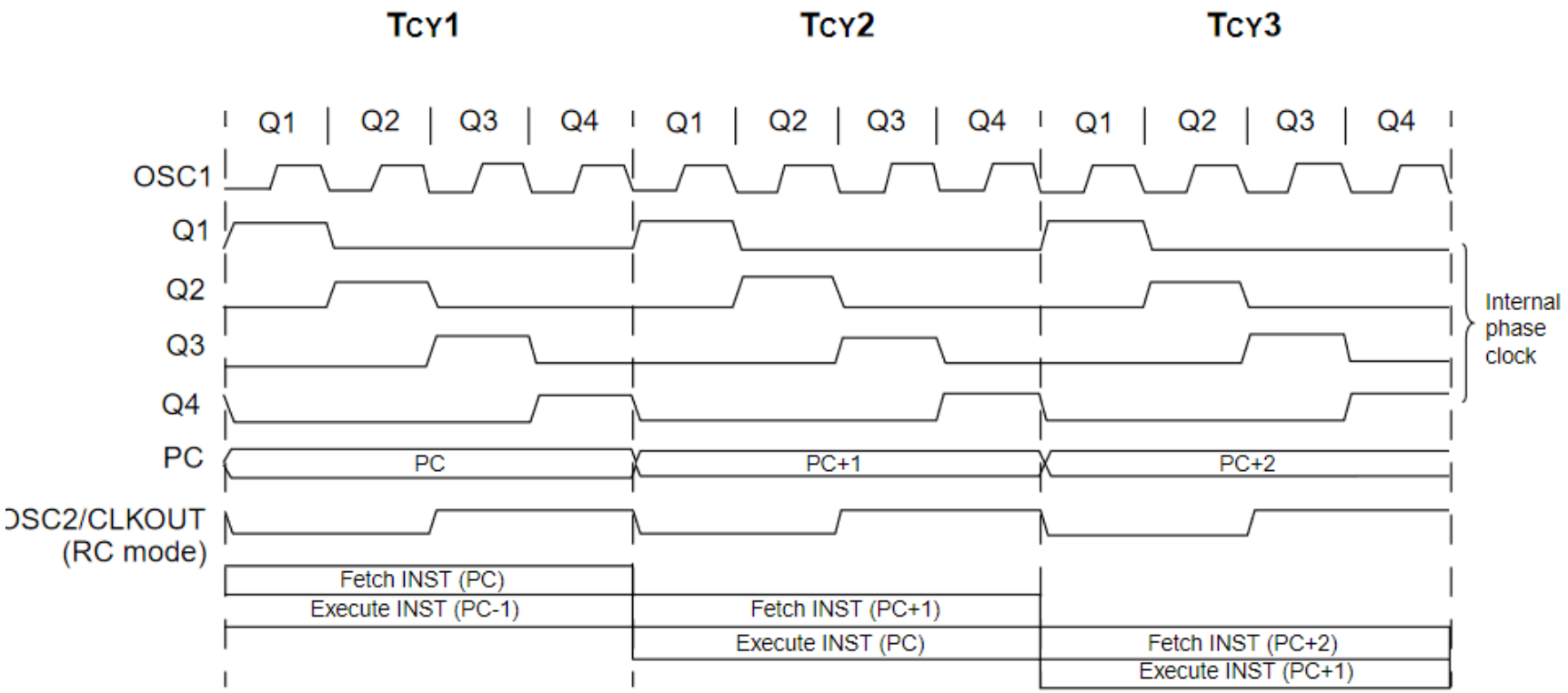
Microchip PIC



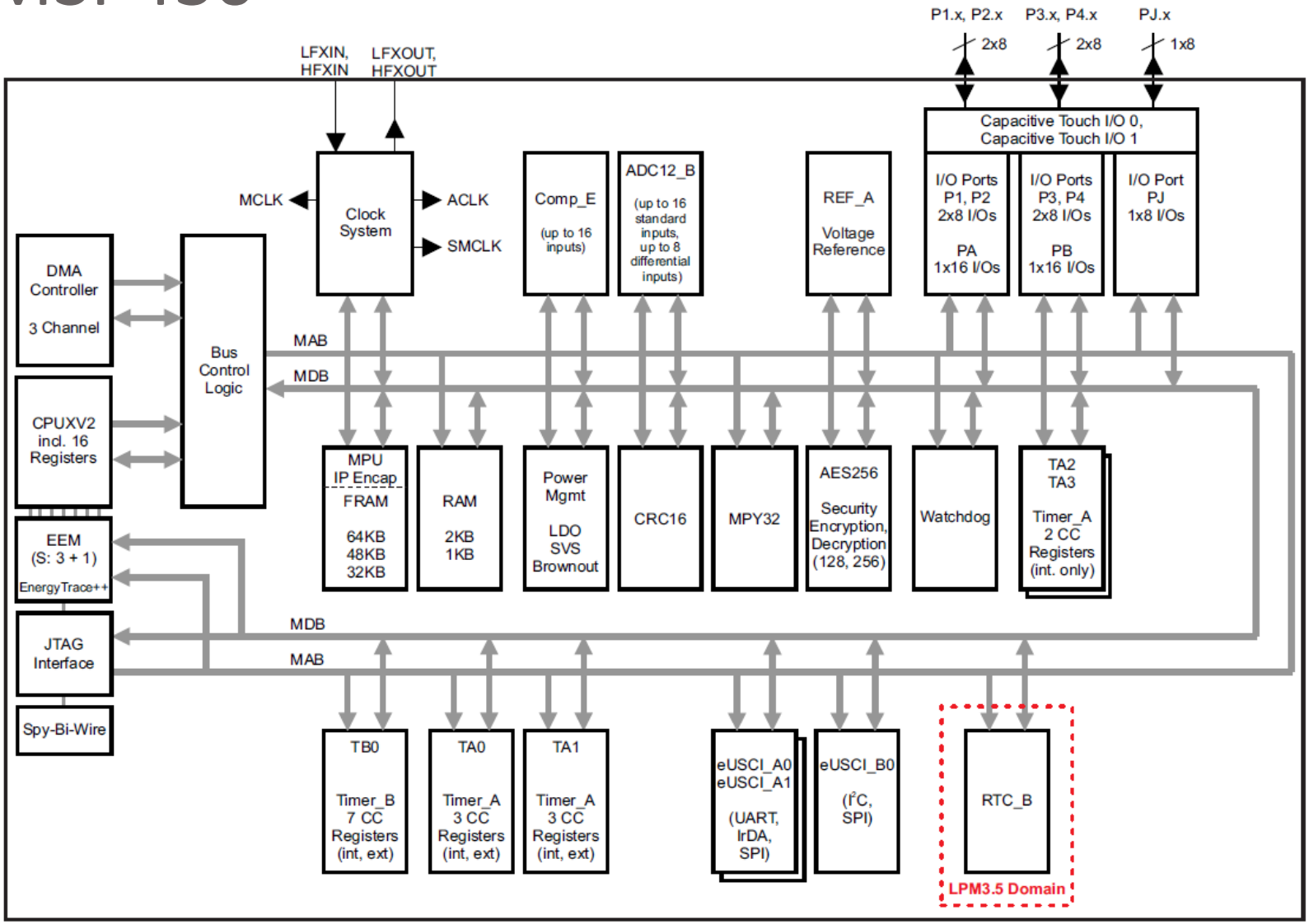
Jedro PIC


Data Bus
PCLATH
PCL
Program Counter
MUX
16-Level Stack
Address Latch
Program Memory
Data Latch
STKPTR
Data Latch
Data Memory
Address Latch
Data Address
BSR
FSR0
FSR1
inc/dec logic
Instruction Bus
Instruction Latch
Address Decode
Data Bus
Instruction Decode and Control
State Machine Control Signals
STATUS Register
MUX
ALU
W Register

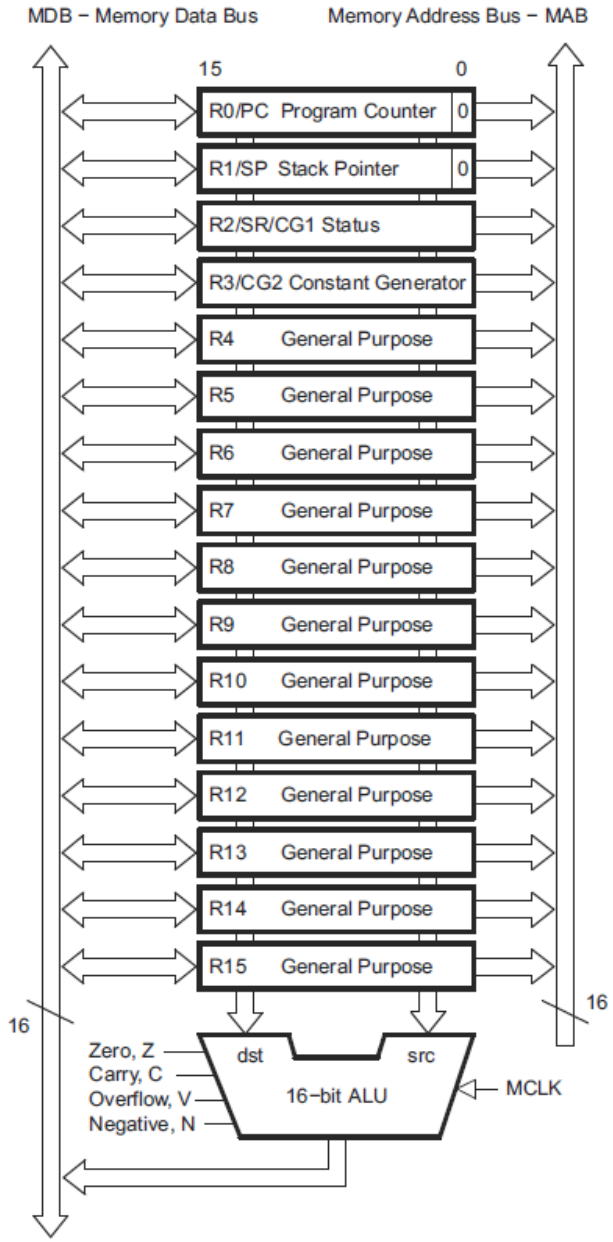
Izvajanje inštrukcij



TI MSP430

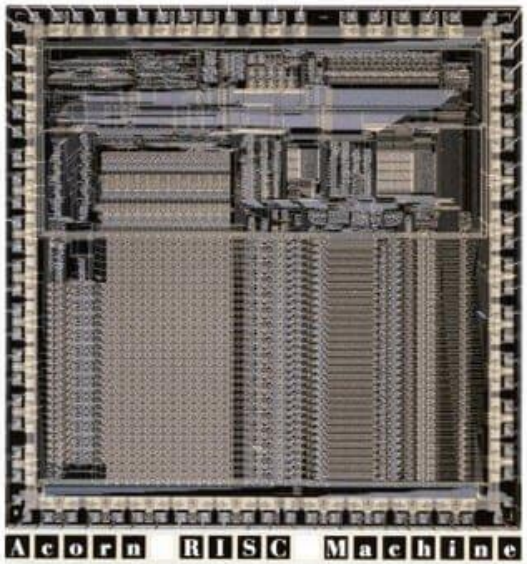

P1.x, P2.x
P3.x, P4.x
PJ.x
2x8
2x8
1x8
LFXIN, HFXIN
LFXOUT, HFXOUT
Capacitive Touch I/O 0, Capacitive Touch I/O 1
MCLK
Clock System
ACLK
SMCLK
Comp_E
(up to 16 inputs)
ADC12_B
(up to 16 standard inputs, up to 8 differential inputs)
REF_A
Voltage Reference
I/O Ports P1, P2 2x8 I/Os
PA 1x16 I/Os
I/O Ports P3, P4 2x8 I/Os
PB 1x16 I/Os
I/O Port PJ 1x8 I/Os
DMA Controller
3 Channel
Bus Control Logic
MAB
MDB
CPUXV2 incl. 16 Registers
MPU IP Encap
FRAM
64KB
48KB
32KB
RAM
2KB
1KB
Power Mgmt
LDO
SVS
Brownout
CRC16
MPY32
AES256
Security Encryption, Decryption (128, 256)
Watchdog
TA2
TA3
Timer_A
2 CC Registers (int. only)
EEM
(S: 3 + 1)
EnergyTrace++
JTAG Interface
Spy-Bi-Wire
MDB
MAB
TB0
Timer_B
7 CC Registers (int, ext)
TA0
Timer_A
3 CC Registers (int, ext)
TA1
Timer_A
3 CC Registers (int, ext)
eUSCI_A0
eUSCI_A1
(UART, IrDA, SPI)
eUSCI_B0
(I2C, SPI)
RTC_B
LPM3.5 Domain

TI MSP430 jedro

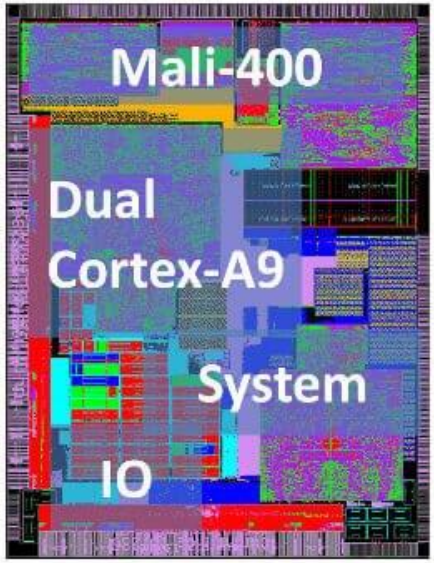


ARM – Advanced RISC Machines

ARM – Advanced RISC Machines



1/10,000th size



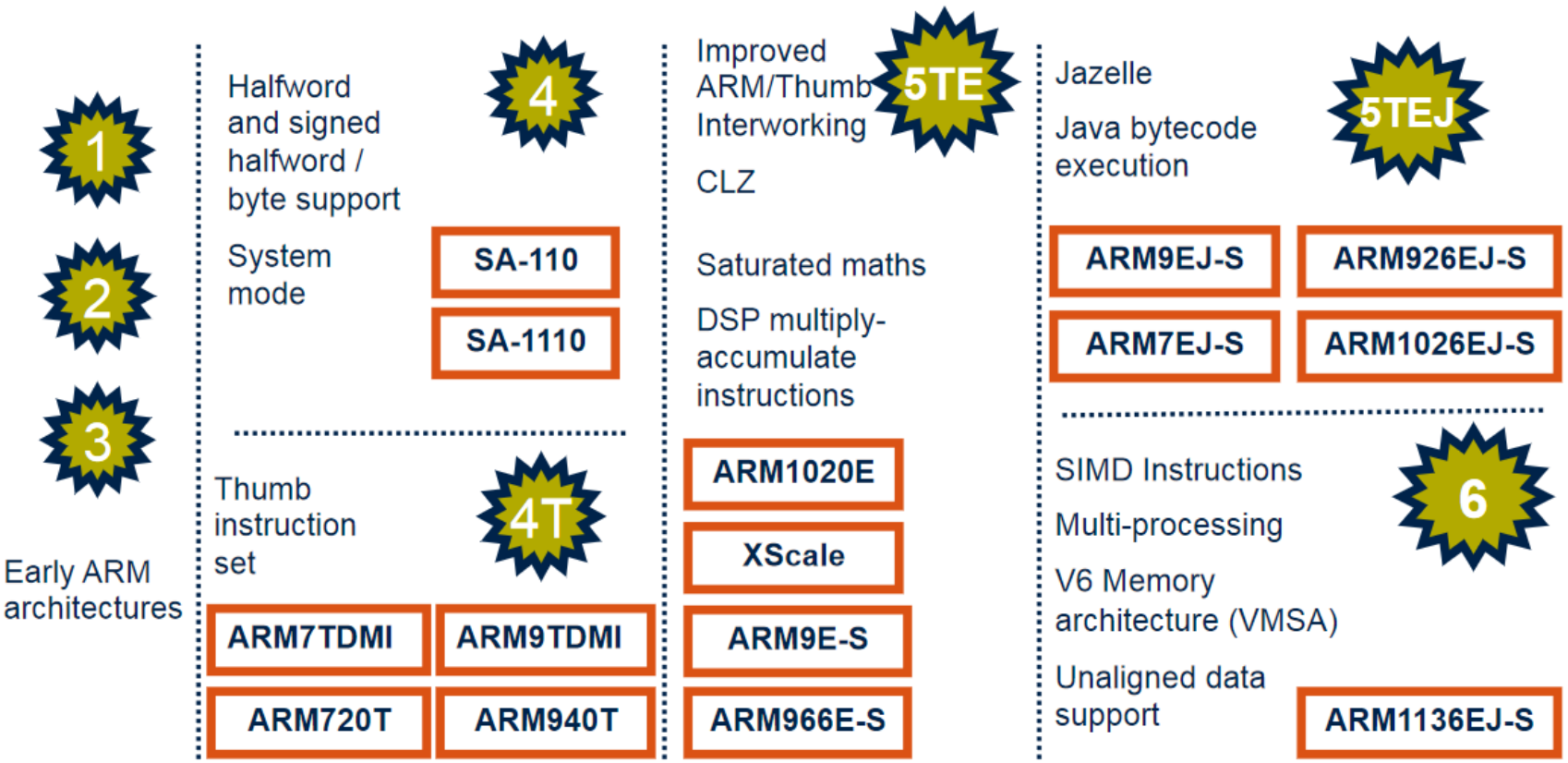
1985 ARM1
3μ 6k gates
7mm x 7mm

2010 Cortex-M0
20nm 8k gates
0.07mm x 0.07mm

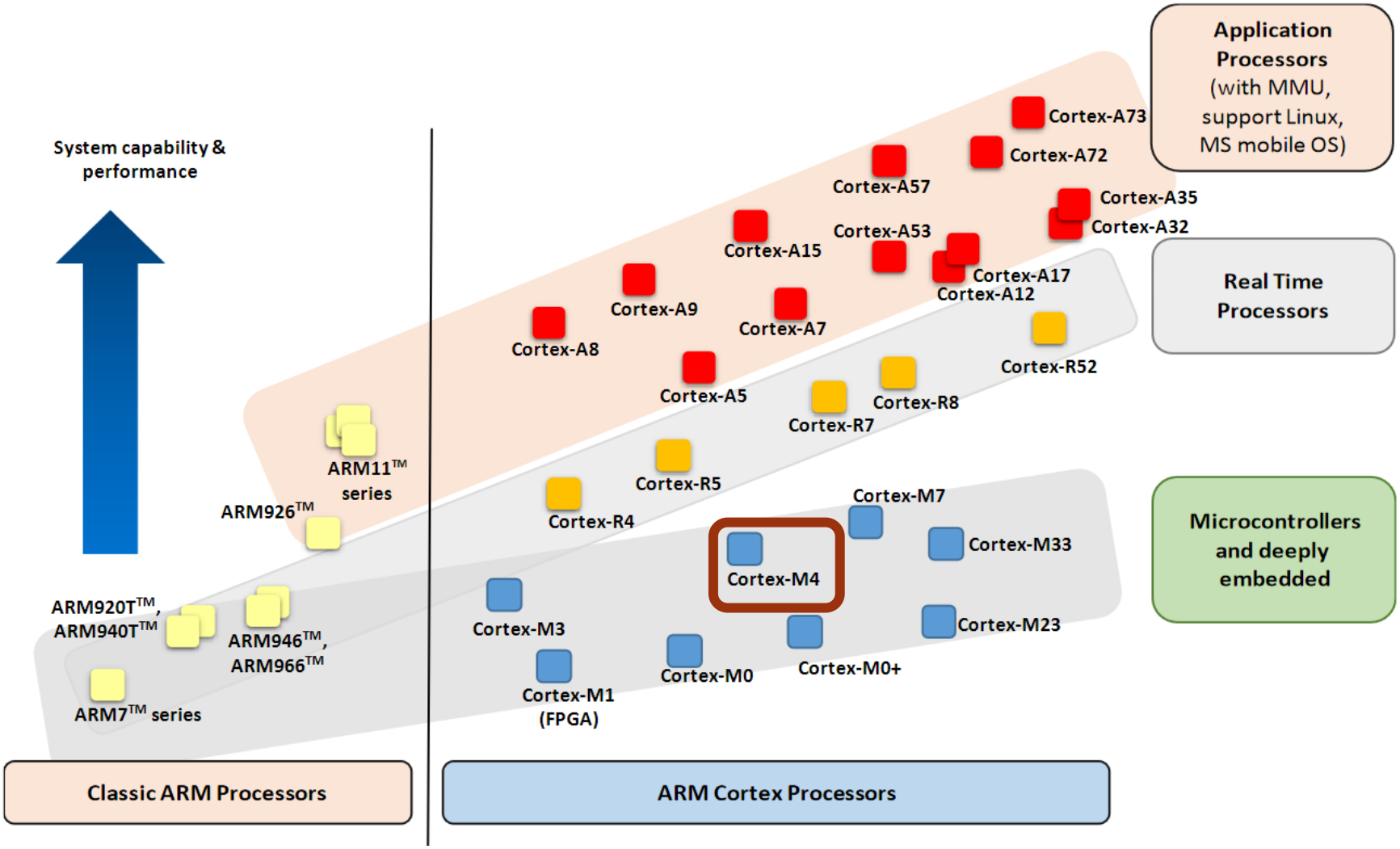
Cortex-A9 SOC
40nm 100M gates
7.4mm x 6.9mm

Acorn computers 1985

Razvoj ARM


1
2
3
Early ARM architectures
Halfword and signed halfword / byte support
4
System mode
SA-110
SA-1110
Thumb instruction set
4T
ARM7TDMI
ARM9TDMI
ARM720T
ARM940T
Improved ARM/Thumb Interworking
5TE
CLZ
Saturated maths
DSP multiply-accumulate instructions
ARM1020E
XScale
ARM9E-S
ARM966E-S
Jazelle
Java bytecode execution
5TEJ
ARM9EJ-S
ARM926EJ-S
ARM7EJ-S
ARM1026EJ-S
SIMD Instructions
Multi-processing
V6 Memory architecture (VMSA)
6
Unaligned data support
ARM1136EJ-S

ARM danes


Application Processors (with MMU, support Linux, MS mobile OS)
Real Time Processors
Microcontrollers and deeply embedded
System capability & performance
Cortex-A73
Cortex-A72
Cortex-A57
Cortex-A35
Cortex-A32
Cortex-A53
Cortex-A15
Cortex-A17
Cortex-A12
Cortex-A9
Cortex-A7
Cortex-A8
Cortex-R52
Cortex-A5
Cortex-R8
Cortex-R7
Cortex-R5
Cortex-R4
Cortex-M7
Cortex-M33
Cortex-M4
Cortex-M3
Cortex-M23
Cortex-M0+
Cortex-M0
Cortex-M1 (FPGA)
ARM11™ series
ARM926™
ARM920T™, ARM940T™
ARM946™, ARM966™
ARM7™ series
Classic ARM Processors
ARM Cortex Processors

Primerjava

	Application processors	Real-time processors	Microcontroller processors
Design	High clock frequency, Long pipeline, High performance, Multimedia support (NEON instruction set extension)	High clock frequency, Long to medium pipeline length, Deterministic (low interrupt latency)	Usually shorter pipeline, Ultra-low-power, Deterministic (low interrupt latency)
System features	Memory Management Unit (MMU), cache memory, ARM TrustZone® security extension	Memory Protection Unit (MPU), cache memory, Tightly Coupled Memory (TCM)	Memory Protection Unit (MPU), Nested Vectored Interrupt Controller (NVIC), Wakeup Interrupt Controller (WIC), ARM TrustZone® security extension in latest designs.
Target markets	Mobile computing, smart phones, energy-efficient servers, high-end microprocessors	Industrial microcontrollers, automotives, Hard disk controllers, Baseband modem	Microcontrollers, Deeply embedded systems (e.g. sensors, MEMS, mixed signal IC), Internet of Things (IoT)

Primerjava ARM cortex M0 – M4

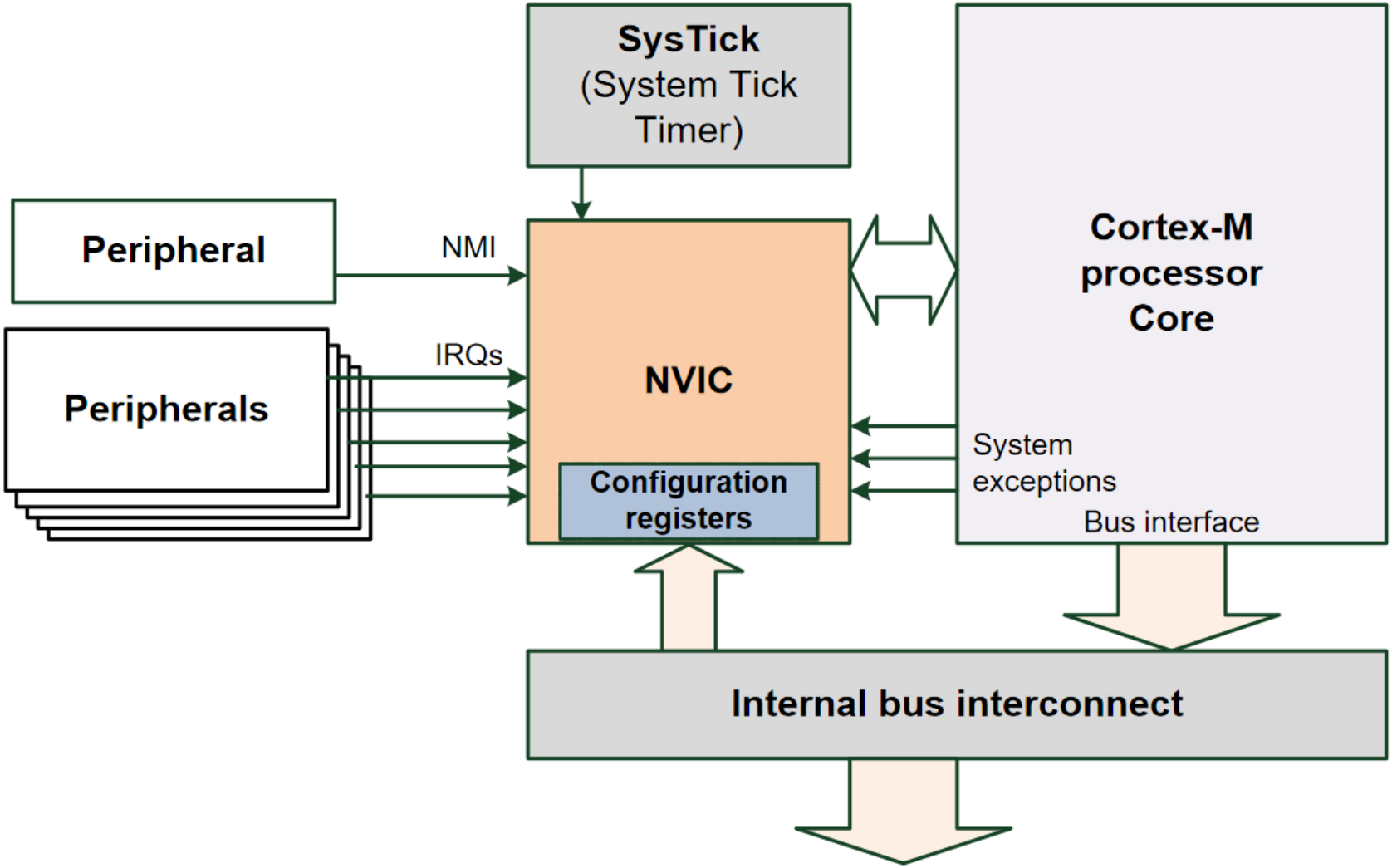
Processor	ARM Architecture	Core Architecture	Thumb®	Thumb®-2	Hardware Multiply	Hardware Divide	Saturated Math	DSP Extensions	Floating Point
Cortex-M0	ARMv6-M	Von Neumann	Most	Subset	1 or 32 cycle	No	No	No	No
Cortex-M0+	ARMv6-M	Von Neumann	Most	Subset	1 or 32 cycle	No	No	No	No
Cortex-M1	ARMv6-M	Von Neumann	Most	Subset	3 or 33 cycle	No	No	No	No
Cortex-M3	ARMv7-M	Harvard	Entire	Entire	1 cycle	Yes	Yes	No	No
Cortex-M4	ARMv7E-M	Harvard	Entire	Entire	1 cycle	Yes	Yes	Yes	Optional

ARM cortex M4 tehnologija

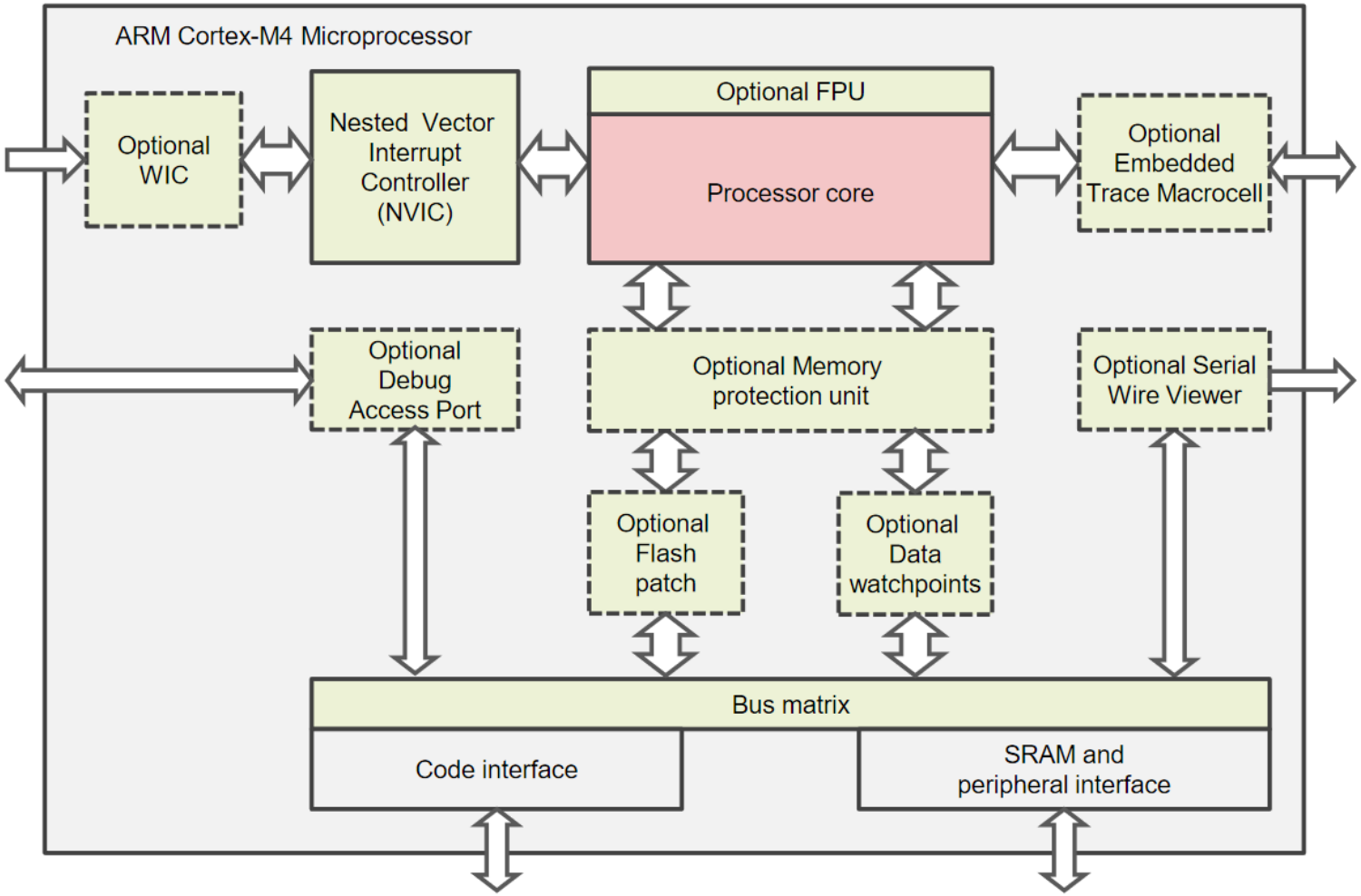
Configuration	90LP (7-track, RVt, typical 1.2V, 25°C)		65LP (8 Track , RVt, typical 1.2V, 25°C)		40LP (9 Track, RVt, typical 1.1v, 25°C)		28HPM (9-track, HVt, typical 0.9v, 85°C)	
	Area mm^2	Power µW/MHz	Area mm^2	Power µW/MHz	Area mm^2	Power µW/MHz	Area mm^2	Power µW/MHz
Minimum Configuration*	0.119	32.82	0.076	26.40	0.028	12.26	0.020	8.47
Feature Rich**	0.304	41.47	0.201	35.54	0.082	15.48	0.053	11.20

Max Freq	40LP (9-track RVt, typical 1.1v, 25C)	28HPM (12track , LVt, typical 0.9v, 85°C)
Feature Rich Configuration**	223MHz	822MHz

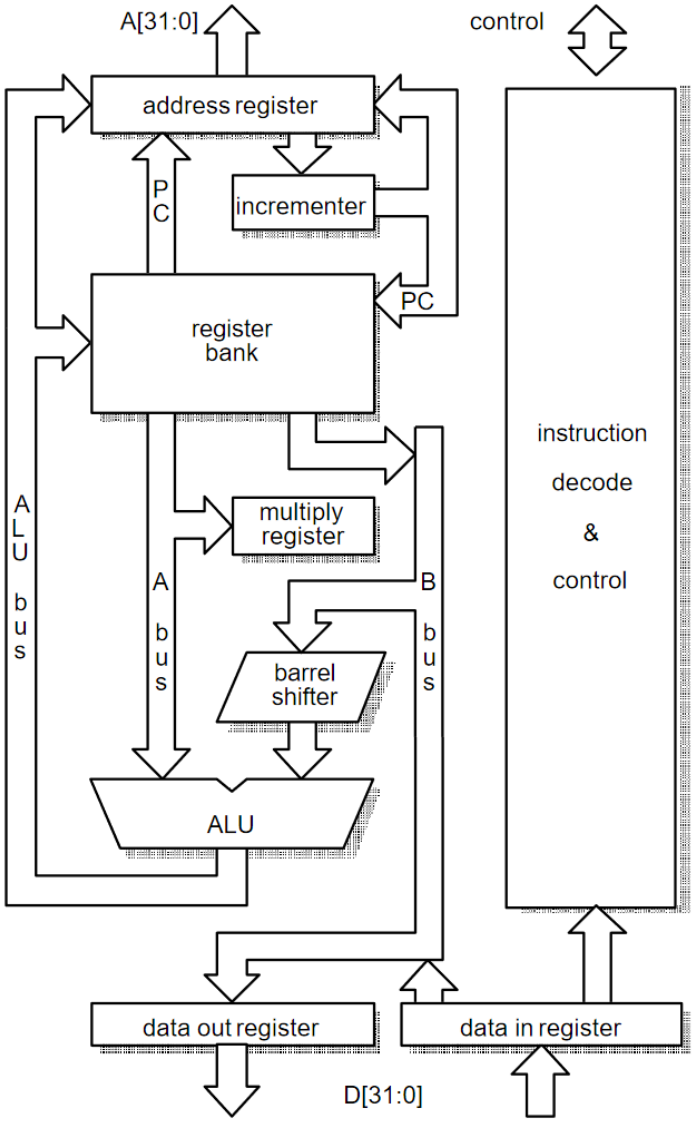
ARM cortex M4 jedro



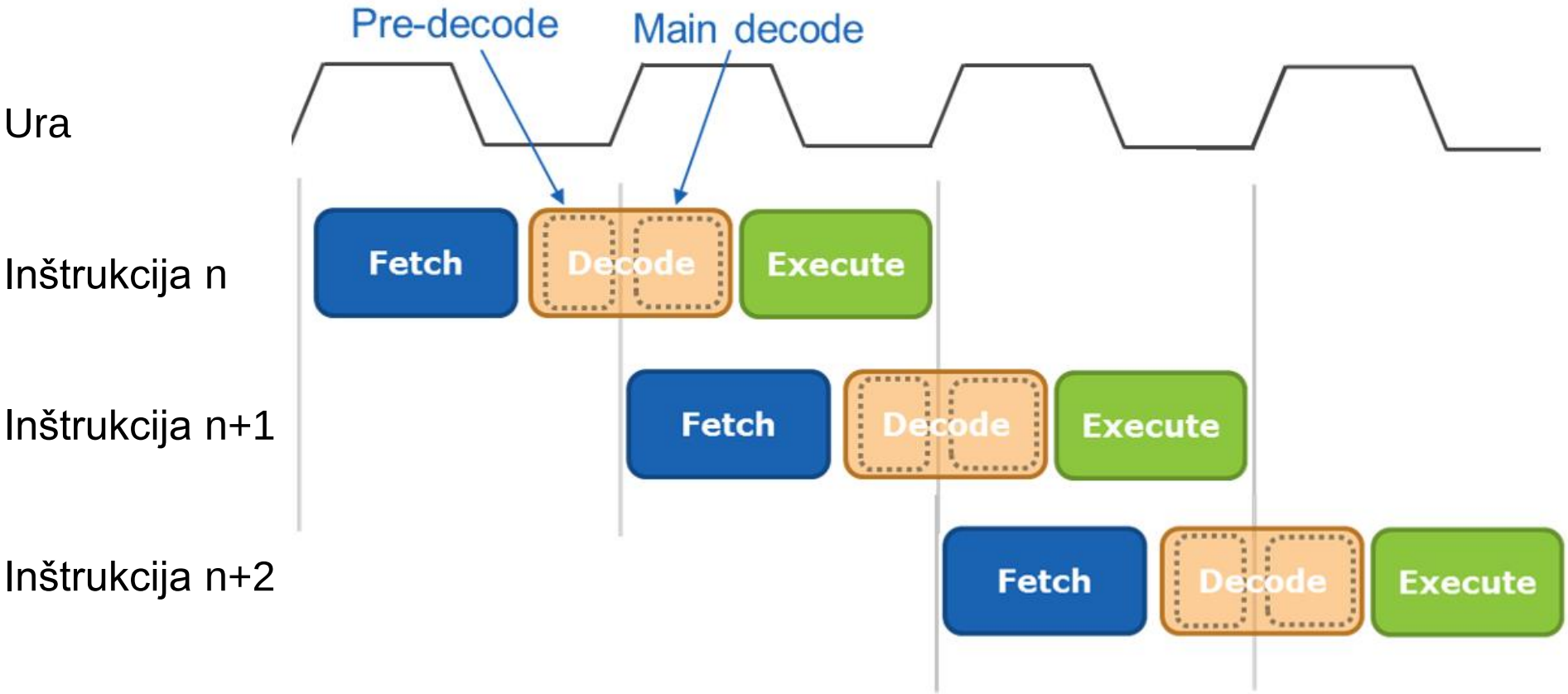
ARM cortex M4 jedro



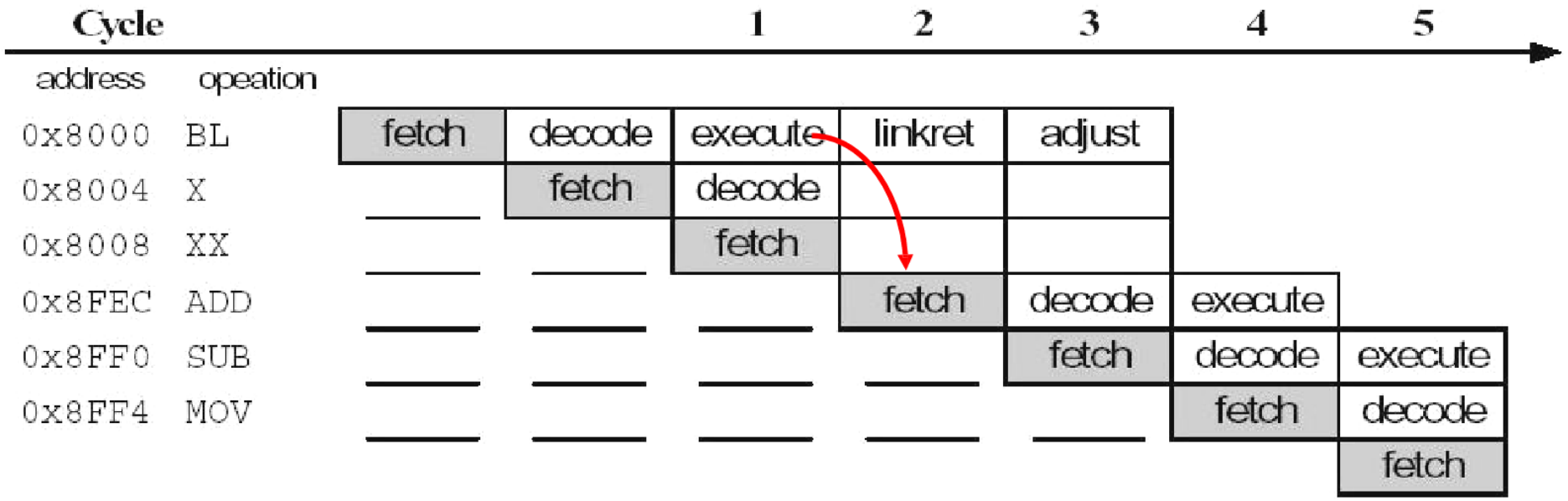
ARM cortex M4 mikroprocesorsko jedro



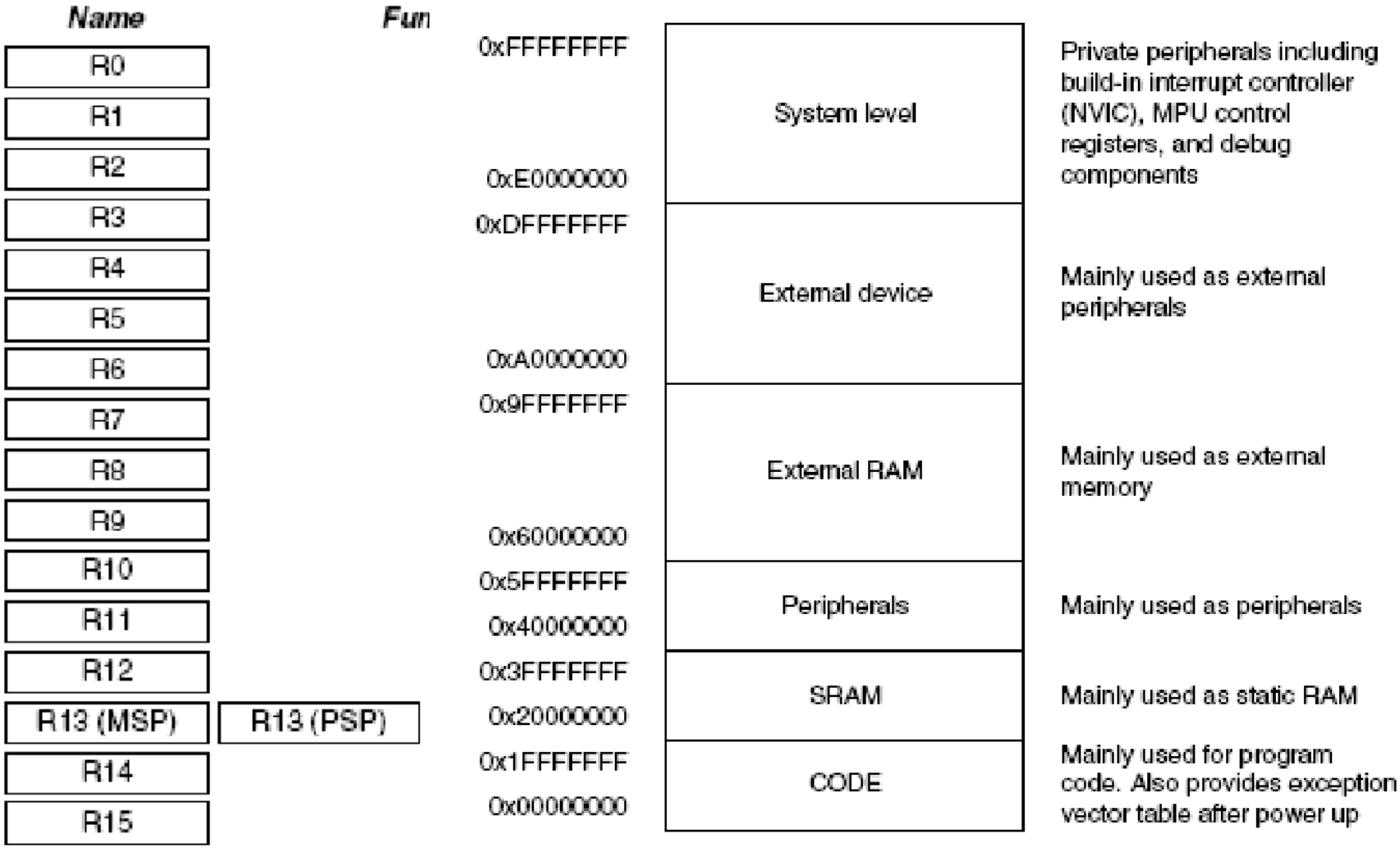
3-nivojsko cevovodenje



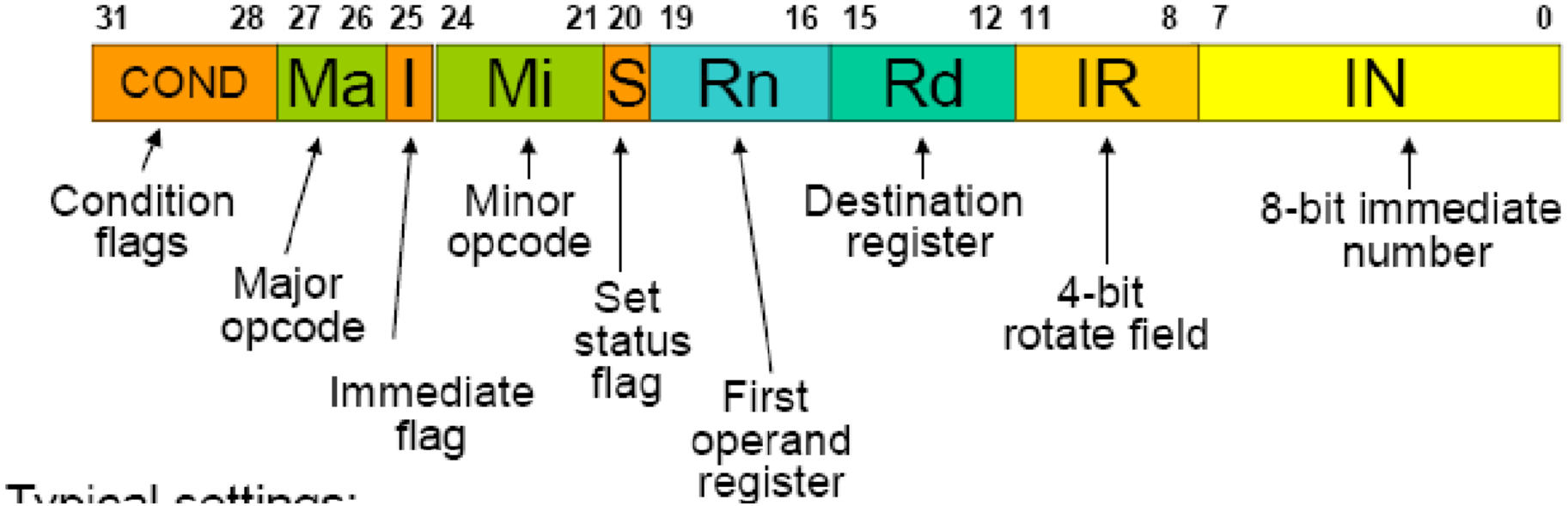
Problem skokov



Registri in pomnilnik



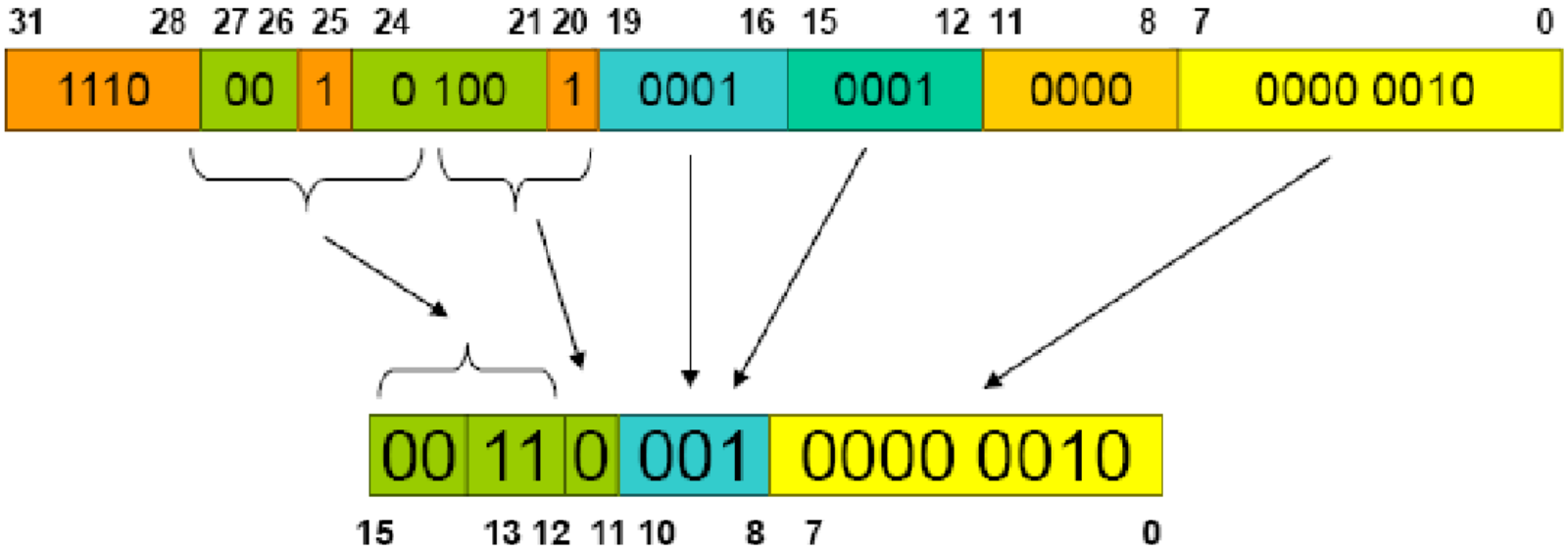
ARM inštrukcije



Typical settings:

Thumb inštrukcije

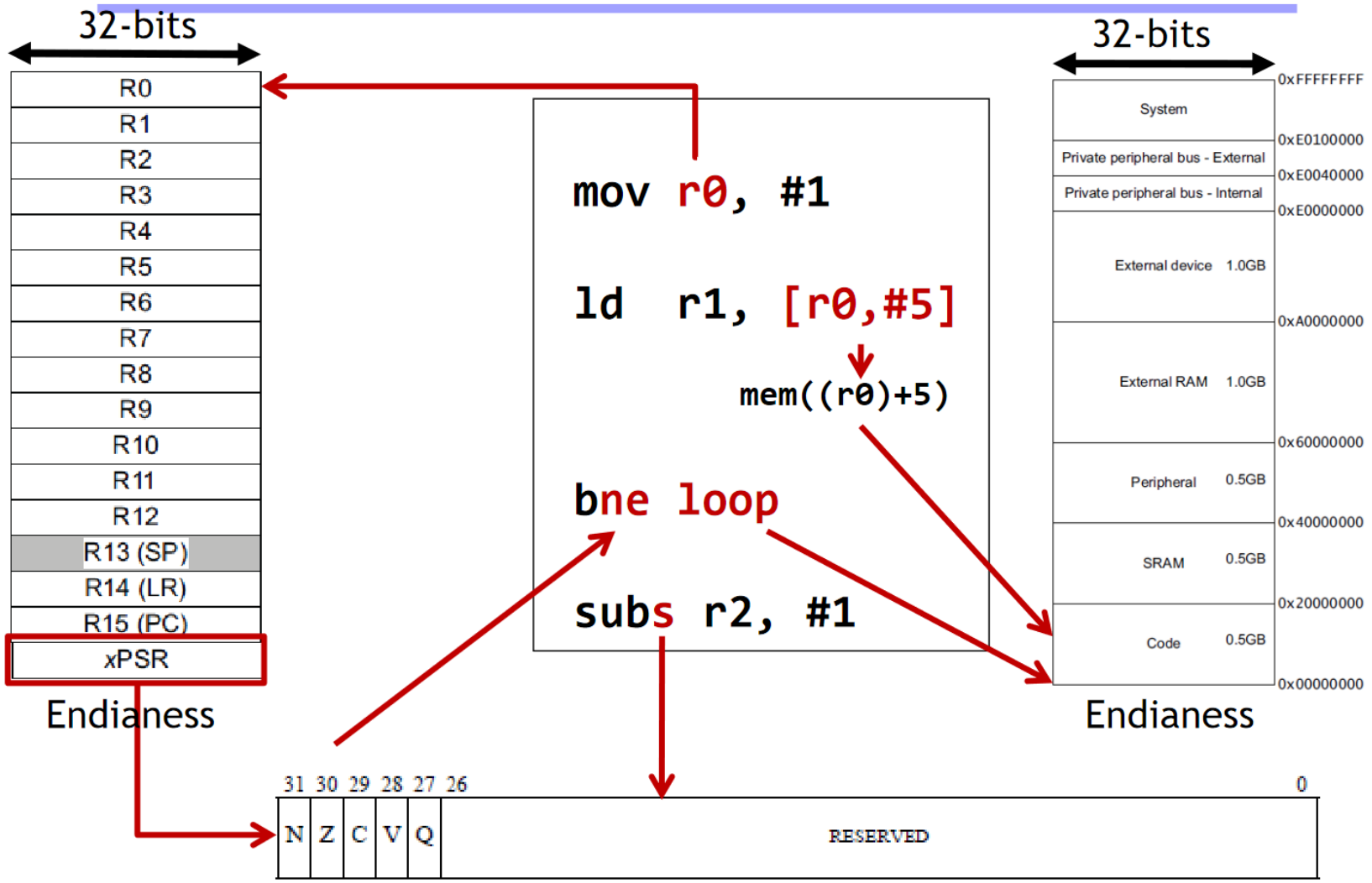
```
ADDS r1, r1, #2
```



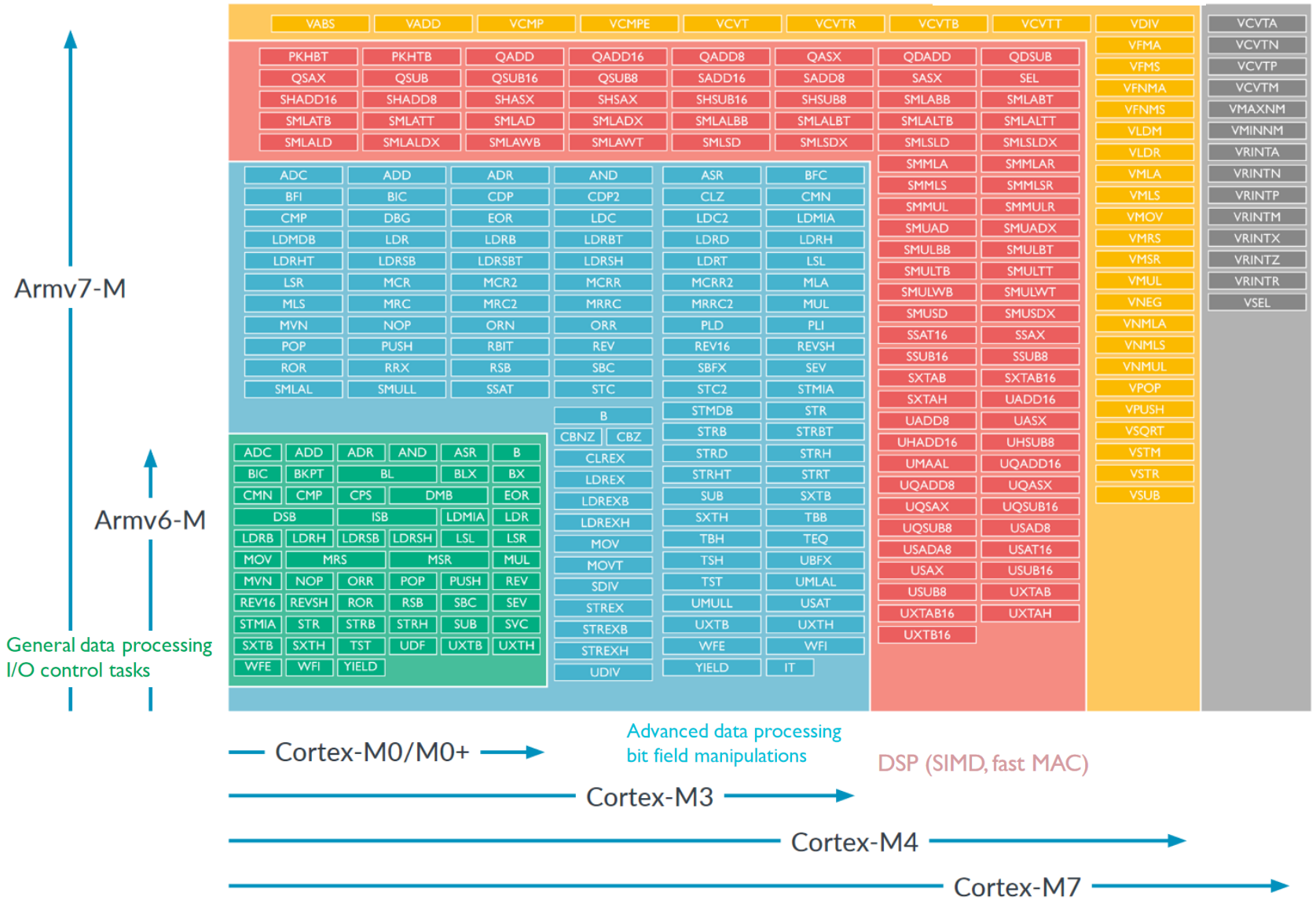

```
ADD r1, #2
```

- Brez pogojev, rotacije, 3-biten naslov registra, ...

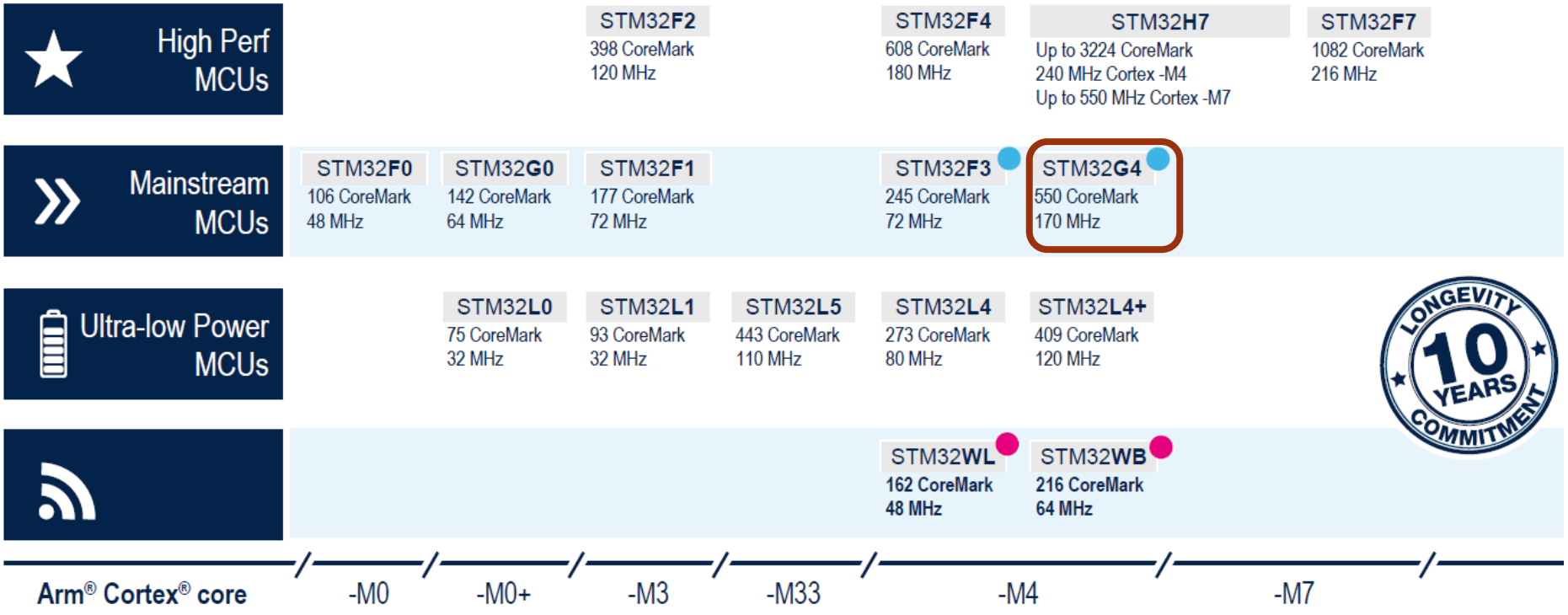
Inštrukcije



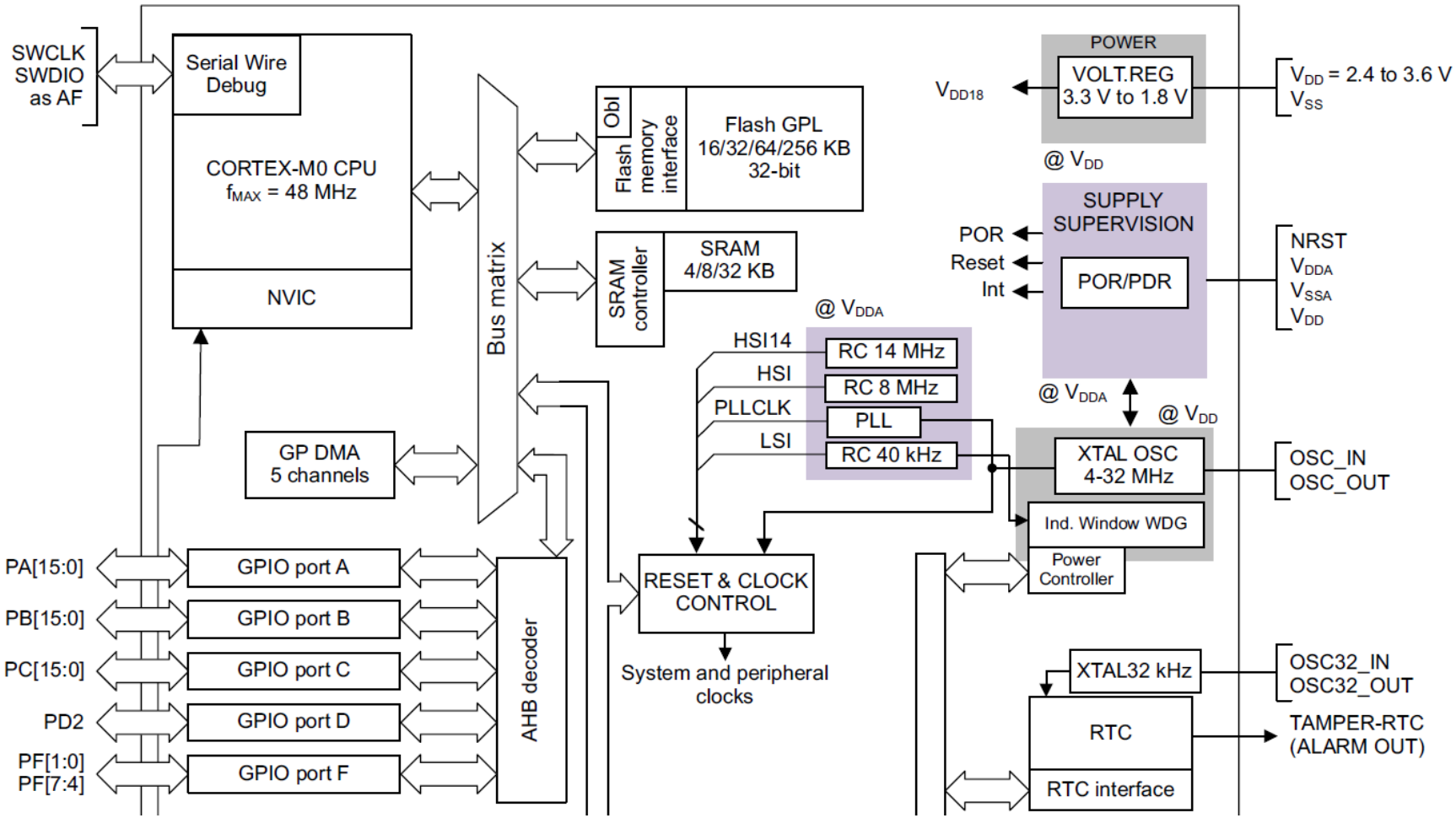
Nabor Thumb inštrukcij



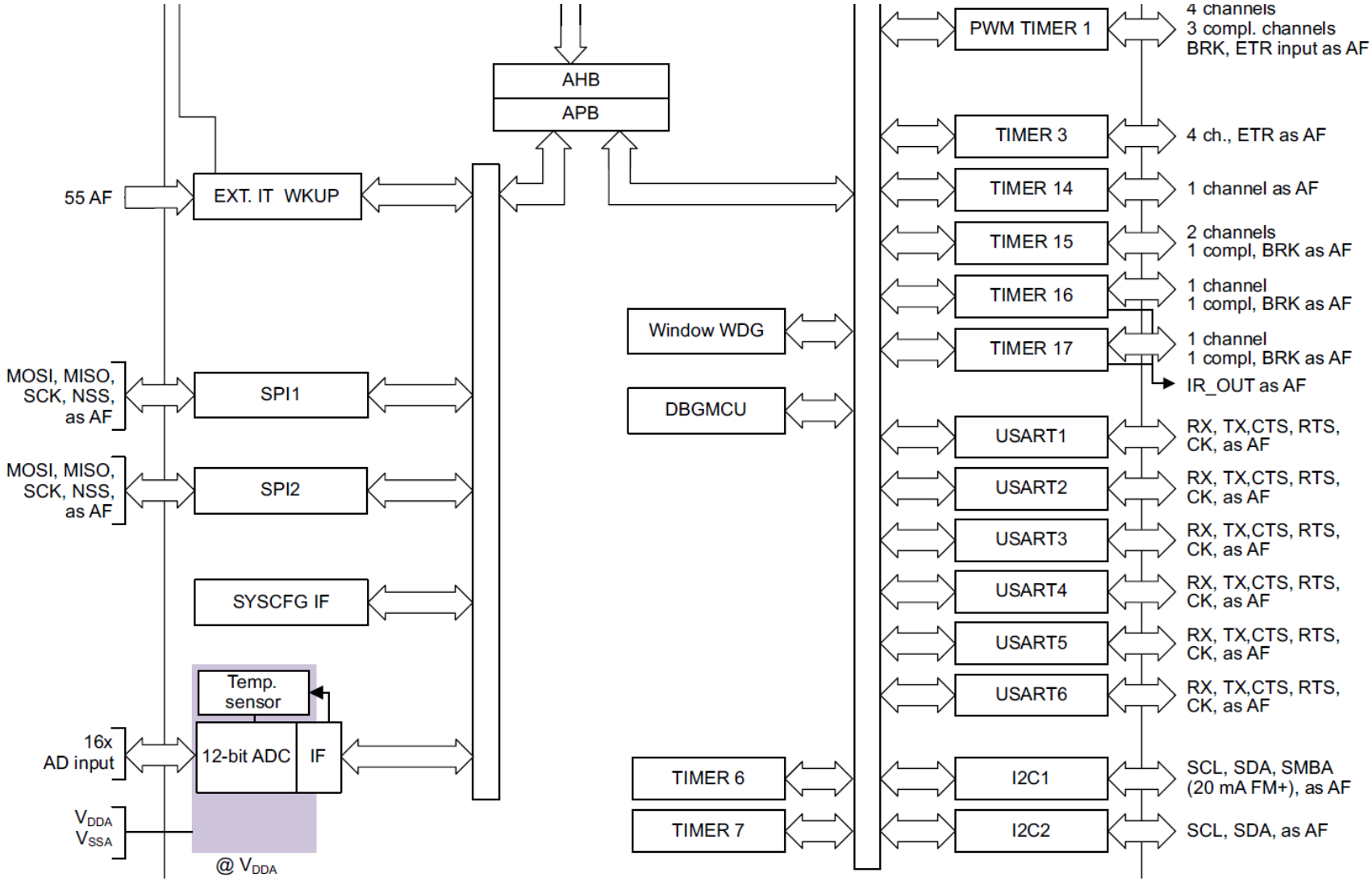
STM32 mikrokrmilniki



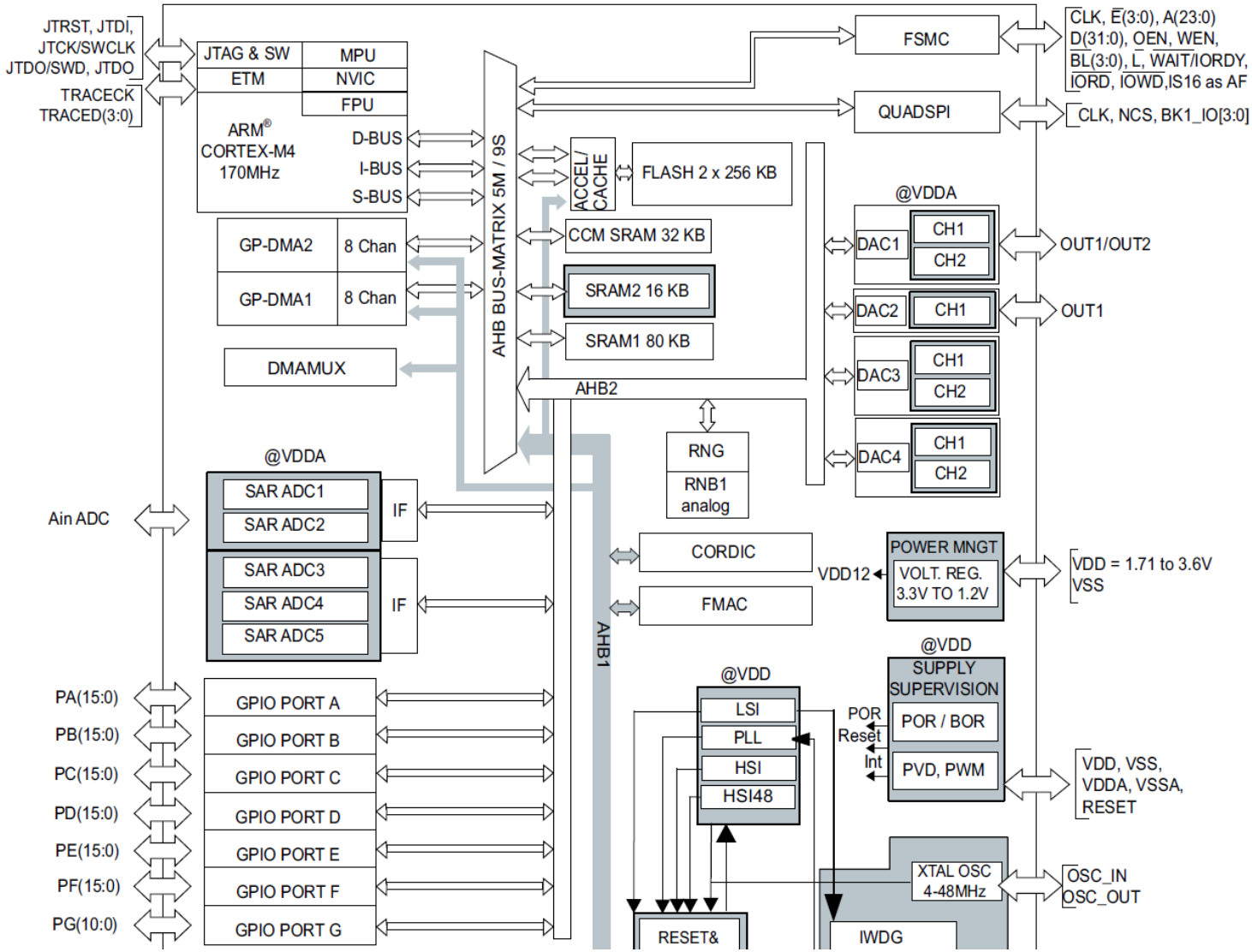
STM32 F0 …


SWCLK
SWDIO
as AF
Serial Wire Debug
CORTEX-M0 CPU
f_{MAX} = 48 MHz
NVIC
Bus matrix
Obl
Flash memory interface
Flash GPL
16/32/64/256 KB
32-bit
SRAM controller
SRAM
4/8/32 KB
GP DMA
5 channels
PA[15:0]
PB[15:0]
PC[15:0]
PD2
PF[1:0]
PF[7:4]
GPIO port A
GPIO port B
GPIO port C
GPIO port D
GPIO port F
AHB decoder
@ V_{DDA}
HSI14
HSI
PLLCLK
LSI
RC 14 MHz
RC 8 MHz
PLL
RC 40 kHz
RESET & CLOCK CONTROL
System and peripheral clocks
POWER
V_{DD18}
VOLT.REG
3.3 V to 1.8 V
@ V_{DD}
V_{DD} = 2.4 to 3.6 V
V_{SS}
SUPPLY SUPERVISION
POR
Reset
Int
POR/PDR
NRST
V_{DDA}
V_{SSA}
V_{DD}
@ V_{DDA}
@ V_{DD}
XTAL OSC
4-32 MHz
OSC_IN
OSC_OUT
Ind. Window WDG
Power Controller
XTAL32 kHz
OSC32_IN
OSC32_OUT
RTC
TAMPER-RTC
(ALARM OUT)
RTC interface

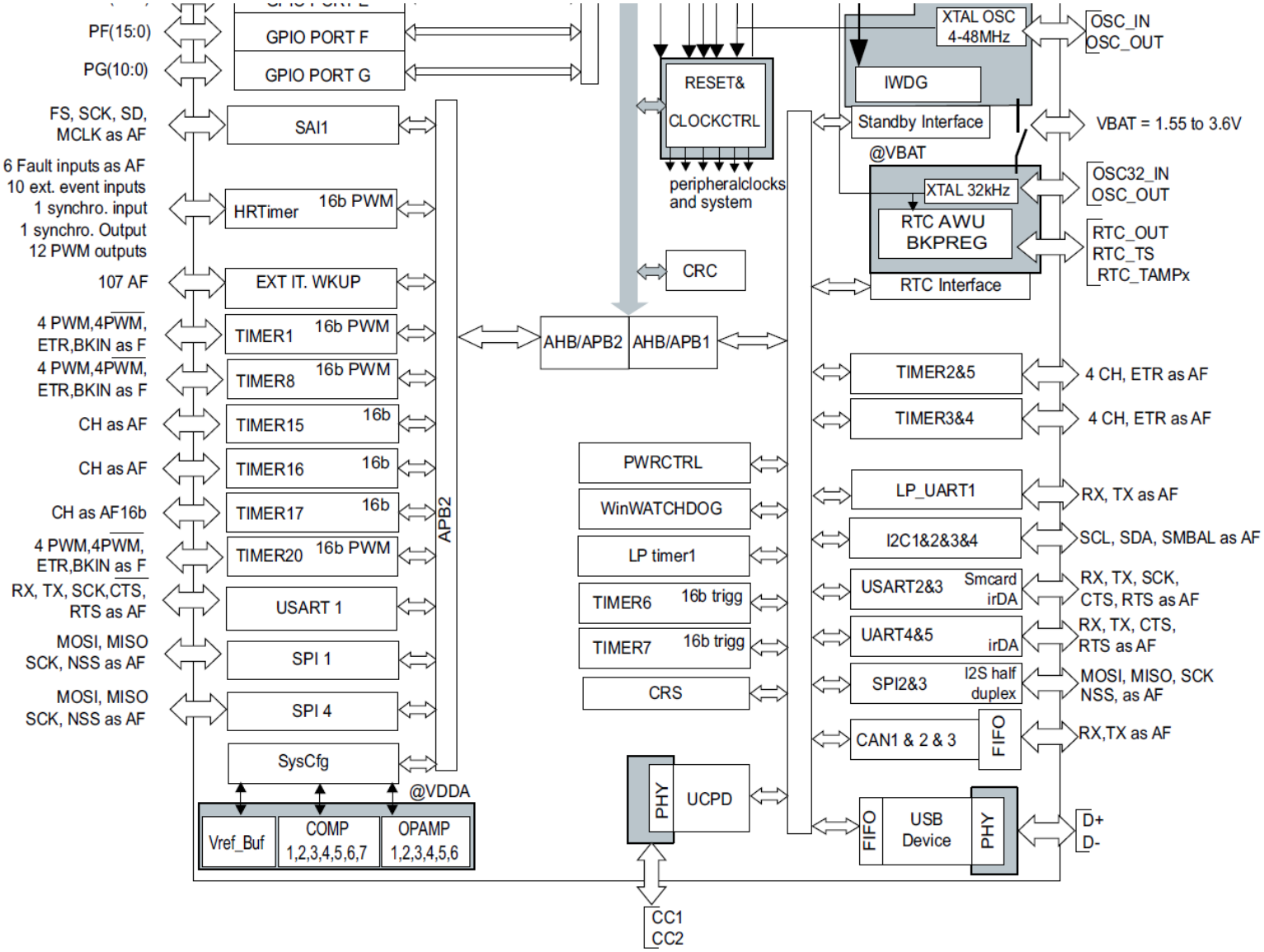
… STM32 F0


AHB
APB
55 AF
EXT. IT WKUP
MOSI, MISO, SCK, NSS, as AF
SPI1
MOSI, MISO, SCK, NSS, as AF
SPI2
SYSCFG IF
Temp. sensor
16x AD input
12-bit ADC
IF
VDDA
VSSA
@ VDDA
Window WDG
DBGMCU
TIMER 6
TIMER 7
PWM TIMER 1
4 channels
3 compl. channels
BRK, ETR input as AF
TIMER 3
4 ch., ETR as AF
TIMER 14
1 channel as AF
TIMER 15
2 channels
1 compl, BRK as AF
TIMER 16
1 channel
1 compl, BRK as AF
TIMER 17
1 channel
1 compl, BRK as AF
IR_OUT as AF
USART1
RX, TX,CTS, RTS, CK, as AF
USART2
RX, TX,CTS, RTS, CK, as AF
USART3
RX, TX,CTS, RTS, CK, as AF
USART4
RX, TX,CTS, RTS, CK, as AF
USART5
RX, TX,CTS, RTS, CK, as AF
USART6
RX, TX,CTS, RTS, CK, as AF
I2C1
SCL, SDA, SMBA (20 mA FM+), as AF
I2C2
SCL, SDA, as AF

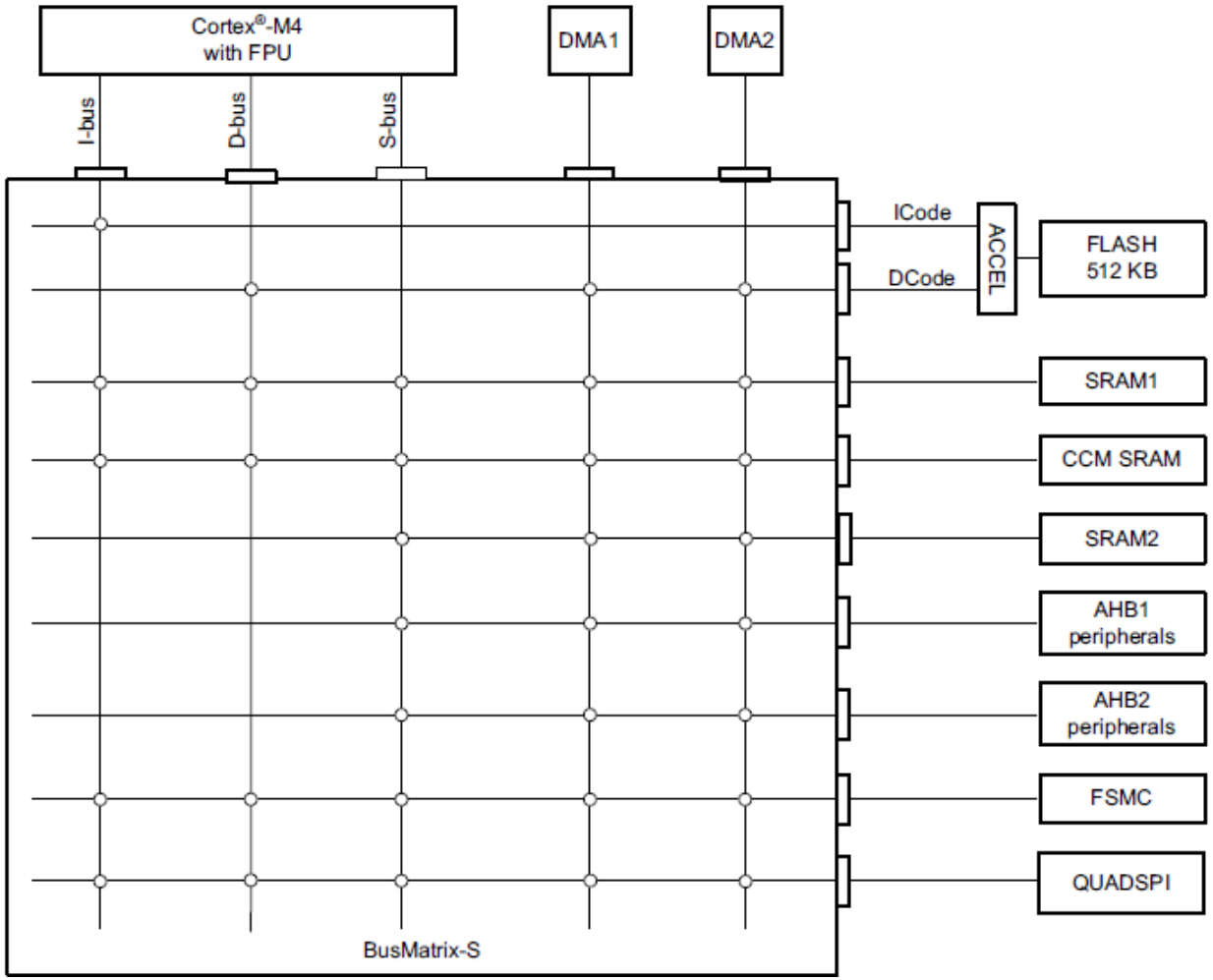
STM32 G4 …


JTRST, JTDI,
JTCK/SWCLK
JTDO/SWD, JTDO
TRACECK
TRACED(3:0)
JTAG & SW
ETM
MPU
NVIC
FPU
ARM®
CORTEX-M4
170MHz
D-BUS
I-BUS
S-BUS
AHB BUS-MATRIX 5M / 9S
ACCEL/
CACHE
FLASH 2 x 256 KB
CCM SRAM 32 KB
SRAM2 16 KB
SRAM1 80 KB
GP-DMA2
8 Chan
GP-DMA1
8 Chan
DMAMUX
AHB2
AHB1
FSMC
QUADSPI
CLK, E(3:0), A(23:0)
D(31:0), OEN, WEN,
BL(3:0), L, WAIT/IORDY,
IORD, IOWD,IS16 as AF
CLK, NCS, BK1_IO[3:0]
@VDDA
DAC1
CH1
CH2
OUT1/OUT2
DAC2
CH1
OUT1
DAC3
CH1
CH2
DAC4
CH1
CH2
RNG
RNB1
analog
@VDDA
SAR ADC1
SAR ADC2
SAR ADC3
SAR ADC4
SAR ADC5
IF
IF
Ain ADC
CORDIC
FMAC
POWER MNGT
VOLT. REG.
3.3V TO 1.2V
VDD12
VDD = 1.71 to 3.6V
VSS
@VDD
SUPPLY
SUPERVISION
POR / BOR
PVD, PWM
POR
Reset
Int
VDD, VSS,
VDDA, VSSA,
RESET
@VDD
LSI
PLL
HSI
HSI48
XTAL OSC
4-48MHz
OSC_IN
OSC_OUT
IWDG
RESET&
PA(15:0)
PB(15:0)
PC(15:0)
PD(15:0)
PE(15:0)
PF(15:0)
PG(10:0)
GPIO PORT A
GPIO PORT B
GPIO PORT C
GPIO PORT D
GPIO PORT E
GPIO PORT F
GPIO PORT G

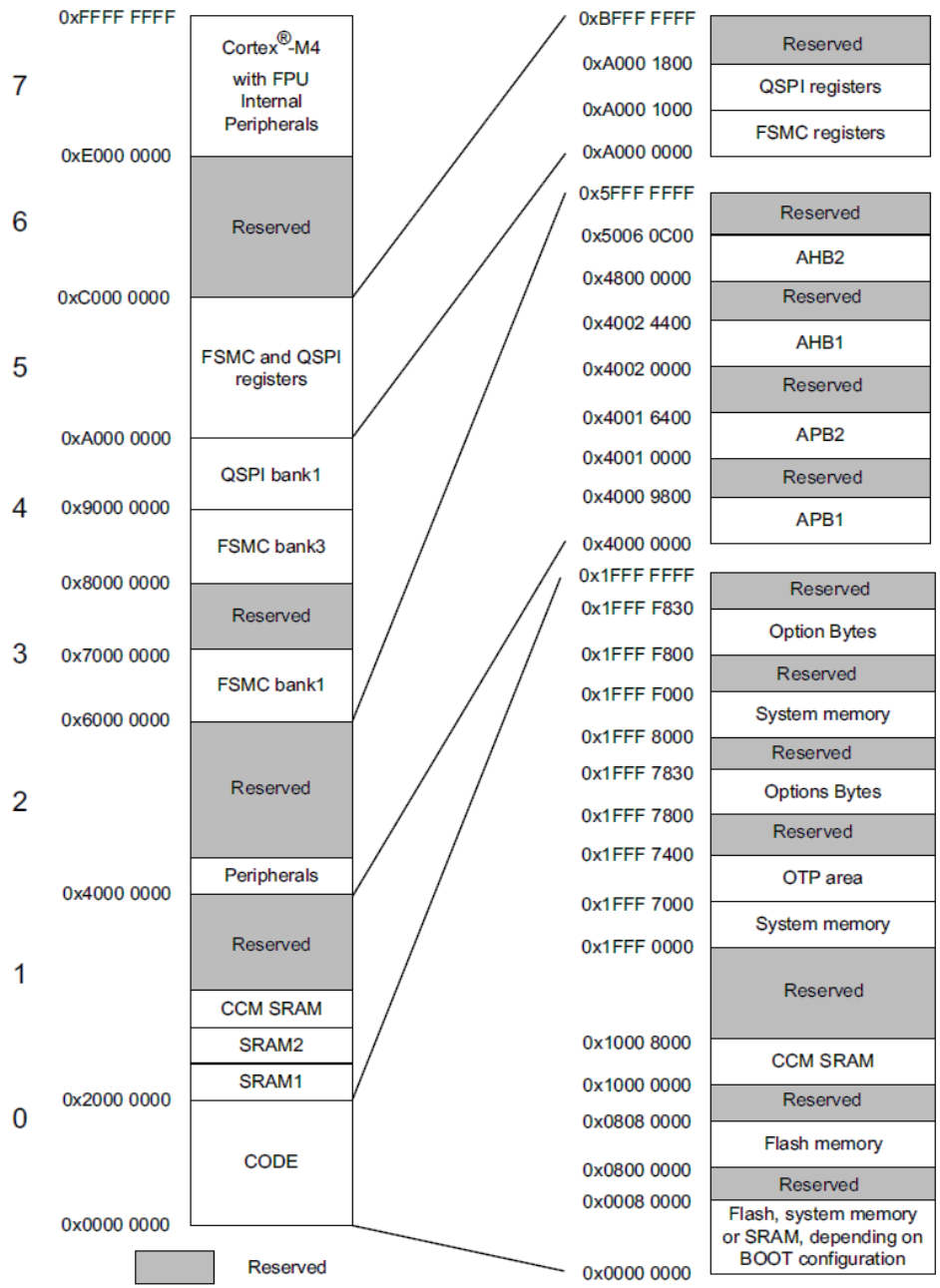
… STM32 G4


PF(15:0)
GPIO PORT F
PG(10:0)
GPIO PORT G
FS, SCK, SD, MCLK as AF
SAI1
6 Fault inputs as AF
10 ext. event inputs
1 synchro. input
1 synchro. Output
12 PWM outputs
HRTimer
16b PWM
107 AF
EXT IT. WKUP
4 PWM,4PWM, ETR,BKIN as F
TIMER1
16b PWM
4 PWM,4PWM, ETR,BKIN as F
TIMER8
16b PWM
CH as AF
TIMER15
16b
CH as AF
TIMER16
16b
CH as AF16b
TIMER17
16b
4 PWM,4PWM, ETR,BKIN as F
TIMER20
16b PWM
RX, TX, SCK,CTS, RTS as AF
USART 1
MOSI, MISO SCK, NSS as AF
SPI 1
MOSI, MISO SCK, NSS as AF
SPI 4
SysCfg
@VDDA
Vref_Buf
COMP 1,2,3,4,5,6,7
OPAMP 1,2,3,4,5,6
APB2
RESET& CLOCKCTRL
peripheralclocks and system
CRC
AHB/APB2
AHB/APB1
PWRCTRL
WinWATCHDOG
LP timer1
TIMER6
16b trigg
TIMER7
16b trigg
CRS
PHY
UCPD
CC1
CC2
XTAL OSC 4-48MHz
OSC_IN
OSC_OUT
IWDG
Standby Interface
@VBAT
VBAT = 1.55 to 3.6V
XTAL 32kHz
OSC32_IN
OSC_OUT
RTC AWU BKPREG
RTC_OUT
RTC_TS
RTC_TAMPx
RTC Interface
TIMER2&5
4 CH, ETR as AF
TIMER3&4
4 CH, ETR as AF
LP_UART1
RX, TX as AF
I2C1&2&3&4
SCL, SDA, SMBAL as AF
USART2&3
Smcard irDA
RX, TX, SCK, CTS, RTS as AF
UART4&5
irDA
RX, TX, CTS, RTS as AF
SPI2&3
I2S half duplex
MOSI, MISO, SCK NSS, as AF
CAN1 & 2 & 3
FIFO
RX,TX as AF
FIFO
USB Device
PHY
D+
D-

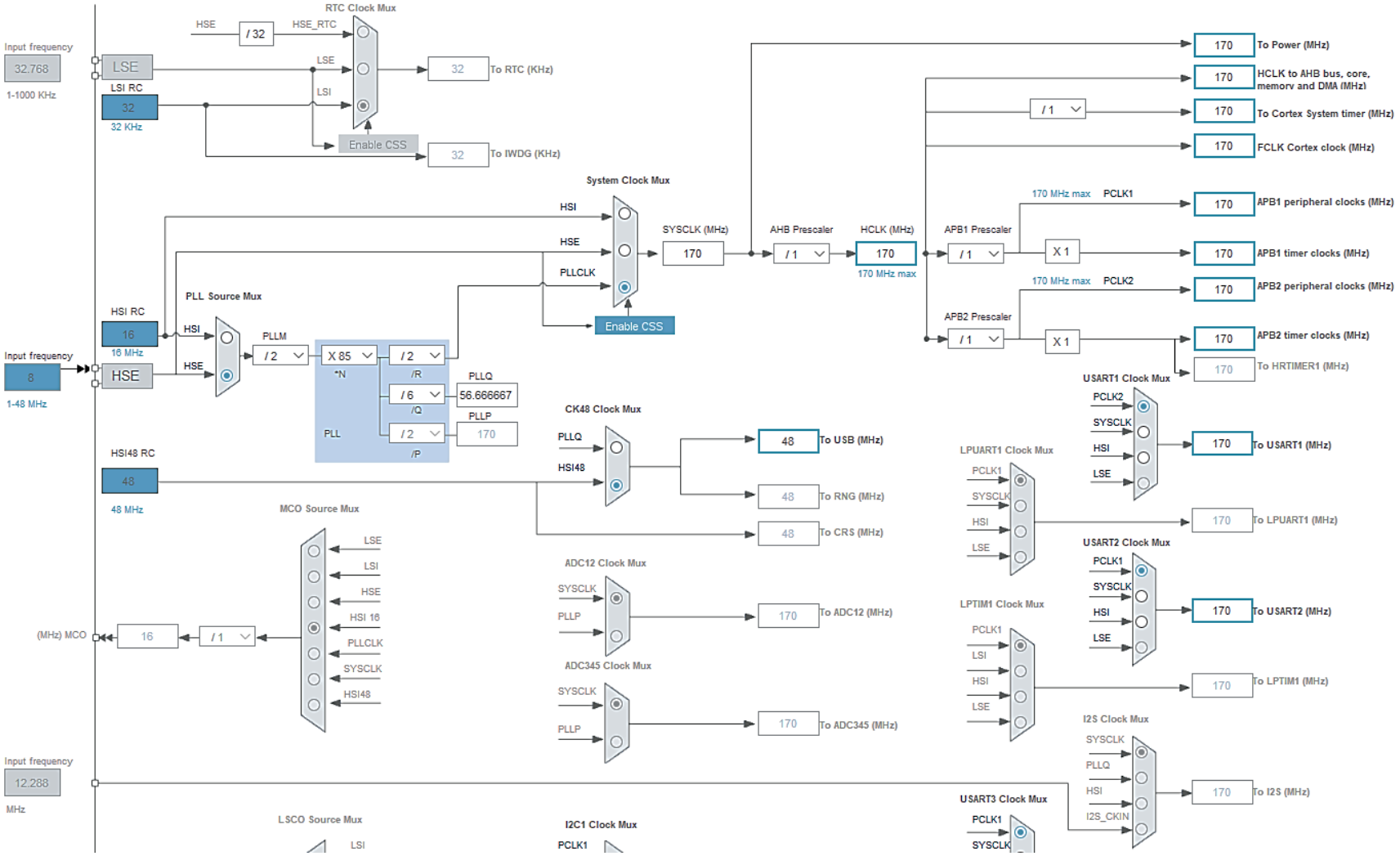
Sistemska arhitektura vodil



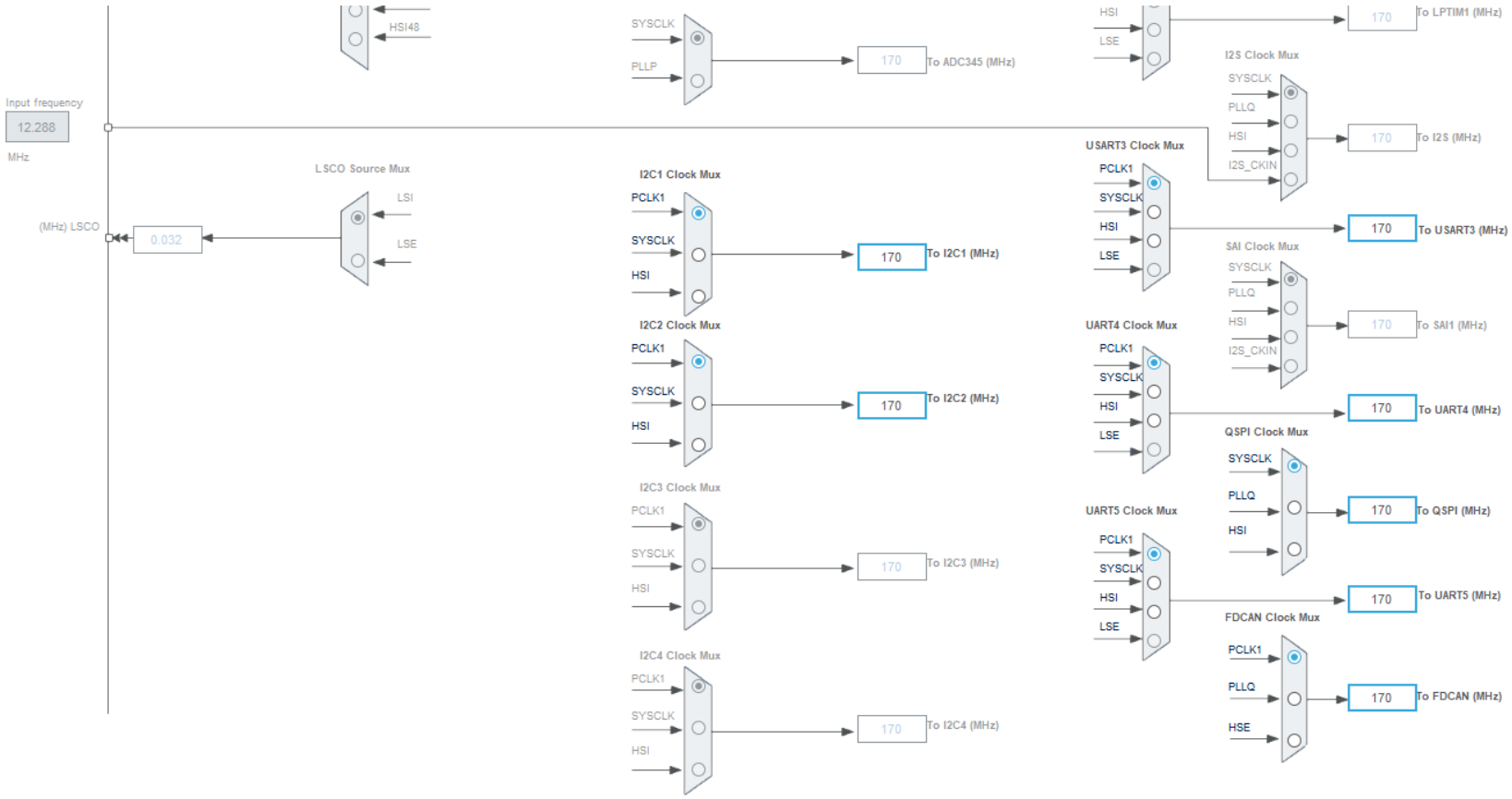
STM32 G4 organizacija pomnilnika



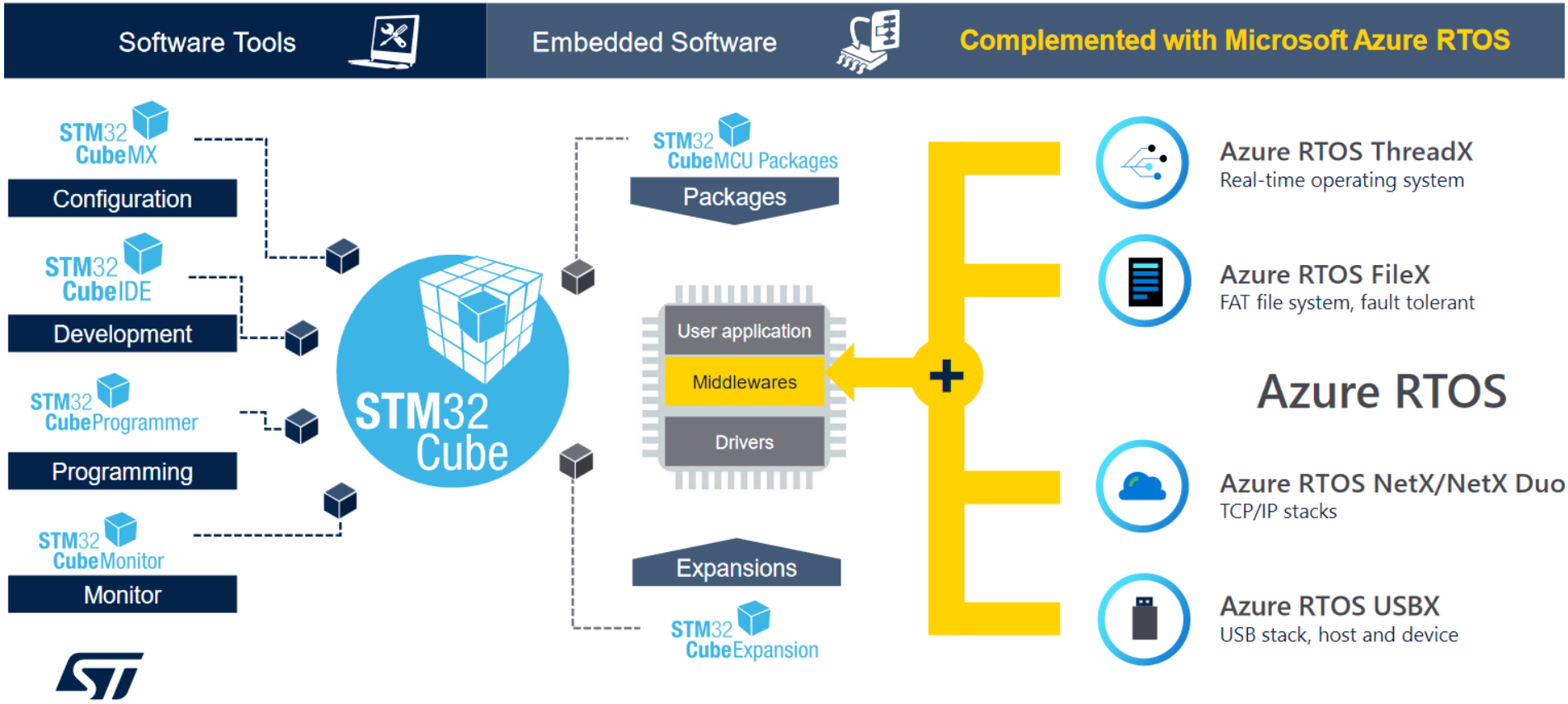
Distribucija ure …



… distribucija ure

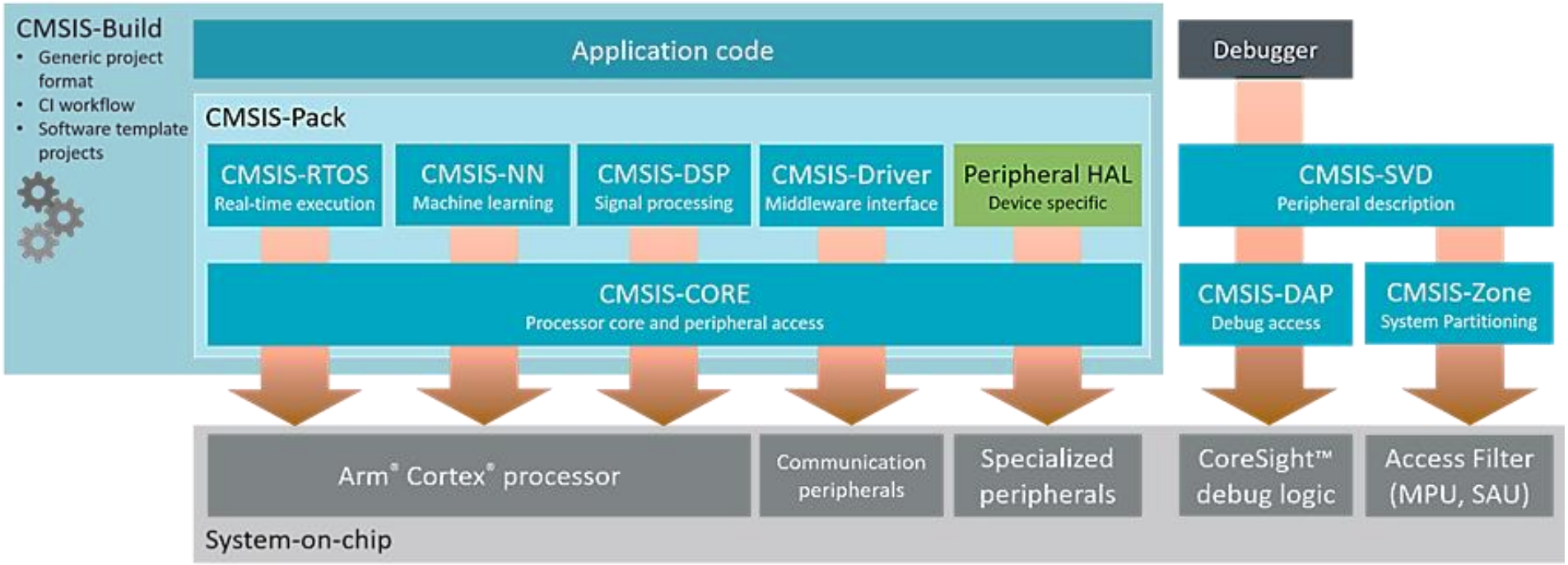


STM32 ekosistem

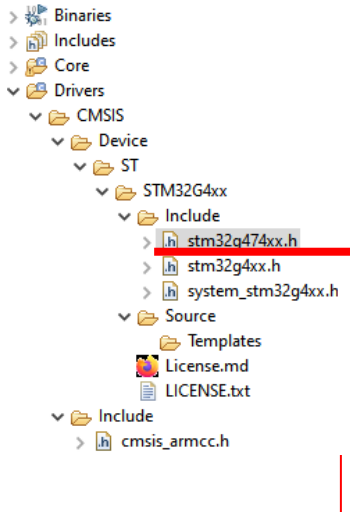


Common Microcontroller Software Interface Standard

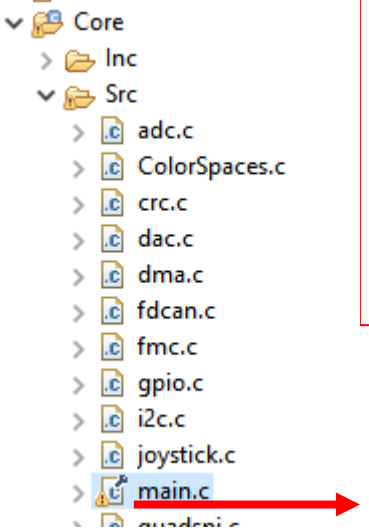
CMSIS



Primer STM32 CMSIS

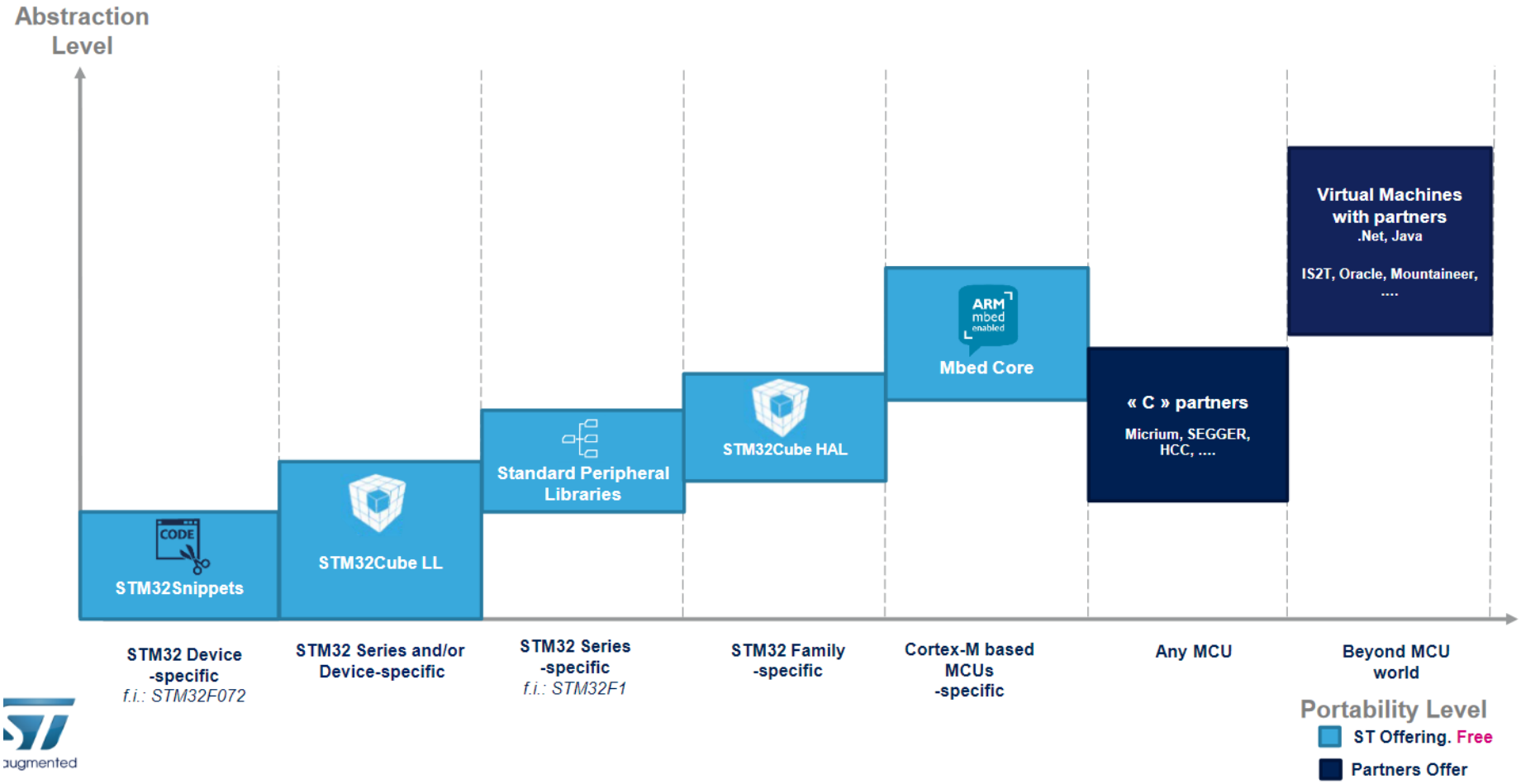



```
typedef struct
{
  __IO uint32_t MODER;       /*!< GPIO port mode register,               Address offset: 0x00      */
  __IO uint32_t OTYPER;      /*!< GPIO port output type register,        Address offset: 0x04      */
  __IO uint32_t OSPEEDR;     /*!< GPIO port output speed register,       Address offset: 0x08      */
  __IO uint32_t PUPDR;       /*!< GPIO port pull-up/pull-down register,  Address offset: 0x0C      */
  __IO uint32_t IDR;         /*!< GPIO port input data register,         Address offset: 0x10      */
  __IO uint32_t ODR;         /*!< GPIO port output data register,        Address offset: 0x14      */
  __IO uint32_t BSRR;        /*!< GPIO port bit set/reset  register,     Address offset: 0x18      */
  __IO uint32_t LCKR;        /*!< GPIO port configuration lock register, Address offset: 0x1C      */
  __IO uint32_t AFR[2];      /*!< GPIO alternate function registers,     Address offset: 0x20-0x24 */
  __IO uint32_t BRR;         /*!< GPIO Bit Reset register,               Address offset: 0x28      */
} GPIO_TypeDef;

.
.
.
#define PERIPH_BASE           (0x40000000UL) /*!< Peripheral base address */
#define GPIOC_BASE            (AHB2PERIPH_BASE + 0x0800UL)
#define AHB2PERIPH_BASE       (PERIPH_BASE + 0x08000000UL)
```

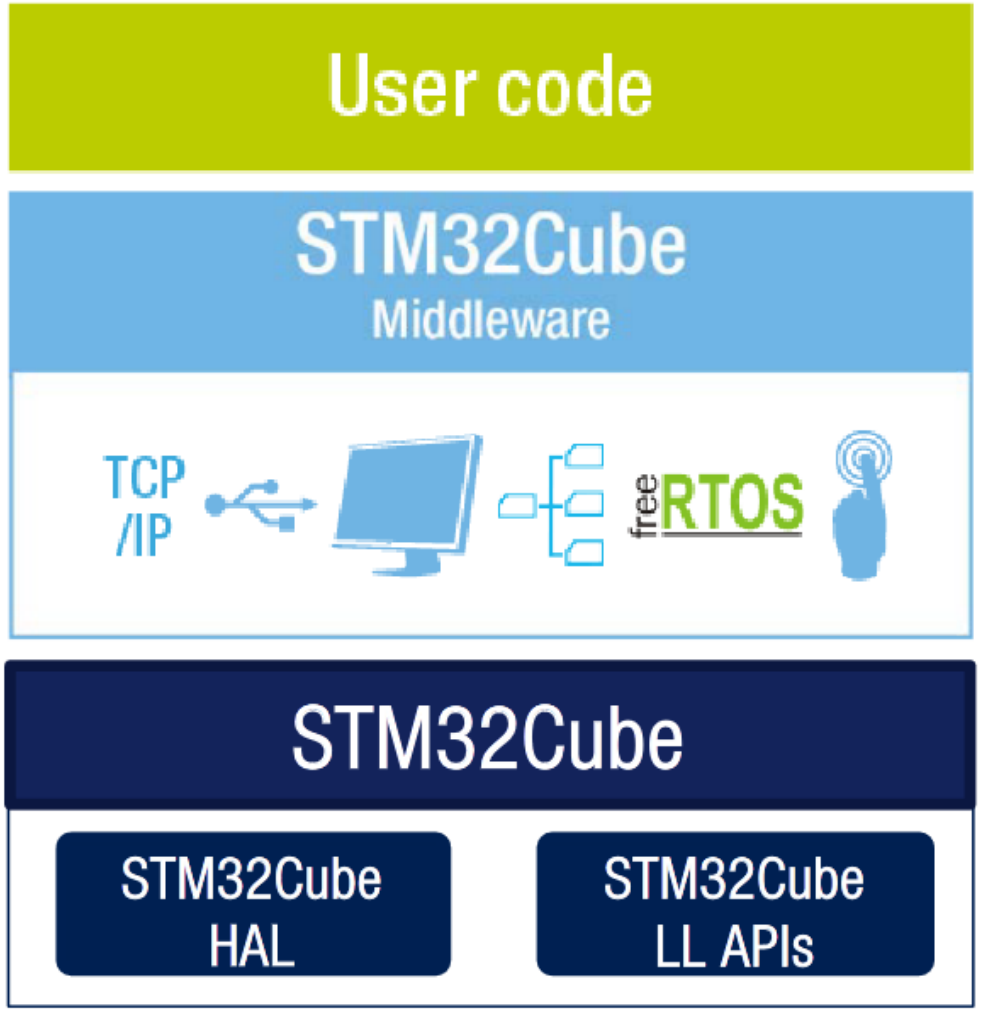



```
#include "stm32g474xx.h"
…
GPIOC->BSRR = 0x00000001;
```

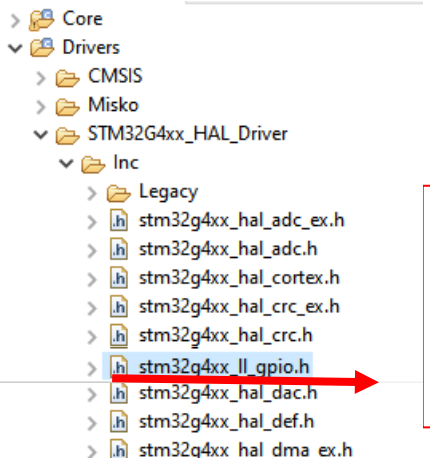
Nivoji knjižnic



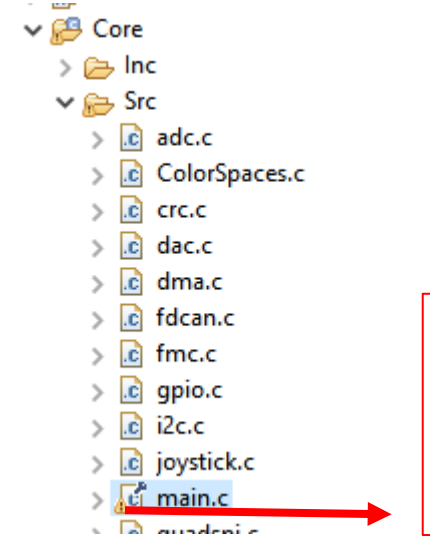
Hardware Abstraction Layer (HAL) Low-Layer (LL)



Primer STM32 LL

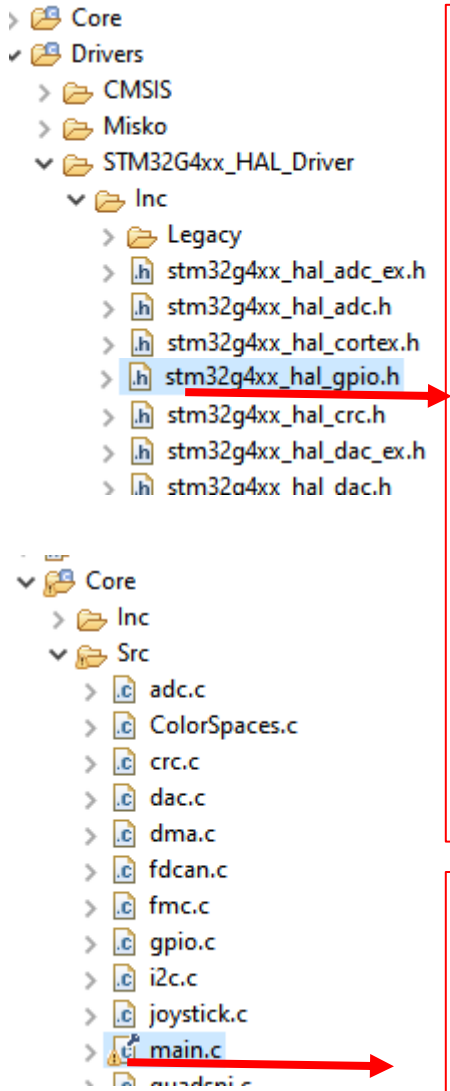



```c
#define WRITE_REG(REG, VAL)   ((REG) = (VAL))

__STATIC_INLINE void LL_GPIO_SetOutputPin(GPIO_TypeDef *GPIOx, uint32_t PinMask)
{
  WRITE_REG(GPIOx->BSRR, PinMask);
}
```




```c
#include “stm32g474xx_ll_gpio.h“

LL_GPIO_SetOutputPin(GPIOC, 1);
```

Primer STM32 HAL




```c
typedef enum
{
  GPIO_PIN_RESET = 0U,
  GPIO_PIN_SET
} GPIO_PinState;

void HAL_GPIO_WritePin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin, GPIO_PinState PinState)
{
  /* Check the parameters */
  assert_param(IS_GPIO_PIN(GPIO_Pin));
  assert_param(IS_GPIO_PIN_ACTION(PinState));

  if (PinState != GPIO_PIN_RESET)
  {
    GPIOx->BSRR = (uint32_t)GPIO_Pin;
  }
  else
  {
    GPIOx->BRR = (uint32_t)GPIO_Pin;
  }
}
}
```

```c
#include “stm32g474xx_hal_gpio.h“

HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, GPIO_PIN_SET);
```