UNIVERSIDAD
DE GRANADA

Sistemas con Microprocesadores

2. Diseño de sistemas con microcontroladores

2.3 Arquitectura de microcontroladores:

ATmega328P

Familia AVR® de Atmel (Microchip)

(https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/8-bit-mcus/avr-mcus)

Características generales

- Familia de microcontroladores desde 1996
- Gama muy amplia de modelos para elección óptima en función de los requisitos de la aplicación
- Amplia difusión gracias a su uso en las plataformas Arduino

Arquitectura AVR

- **Arquitectura RISC tipo Harvard:**
  - Memoria de datos y de programa separadas: instrucciones y datos de distinta longitud (16 y 8 bits respectivamente) y posibilidad de acceso simultáneo mediante buses distintos.
  - Repertorios reducido de instrucciones máquina; todas las instrucciones de una palabra (longitud constante).
  - Cauce segmentado de dos etapas: simultaneidad de ejecución de una instrucción y captación de la siguiente.
- Clasificación en subfamilias con diferentes prestaciones.
- La misma arquitectura implementada con distintos tamaños de memoria y distintos recursos de entradas/salidas.

Subfamilias AVR de 8-bits (*datapath* o camino de datos)

Nombre Ser.	Patillas	Mem. Flash	Características particulares
ATtiny	6-32	0.5-16 KB	Pequeño tamaño
ATmega	28-100	4-256 KB	Periféricos ampliados
ATxmega	44-100	16-384 KB	DMA, Event System, Crypto
AVR DA*	28-64	16-128 KB	Táctil capacit., 12b ADC, 10bDAC
AVR DB*	14-64	16-128 KB	MVIO, CCL, 12b ADC, 10b DAC
AVR DD*	14-32	16-64 KB	MVIO, CCL, USART, ADC, DAC
Apl. específ.	Controlad. LCD, controlad. USB , PWM avanzado, CAN, etc.		

*** Introducido en 2020**
MVIO: Multi Voltage Input/Output
CCL: Configurable Custom Logic
USART: Universal Asynchronous Receiver/Transmitter

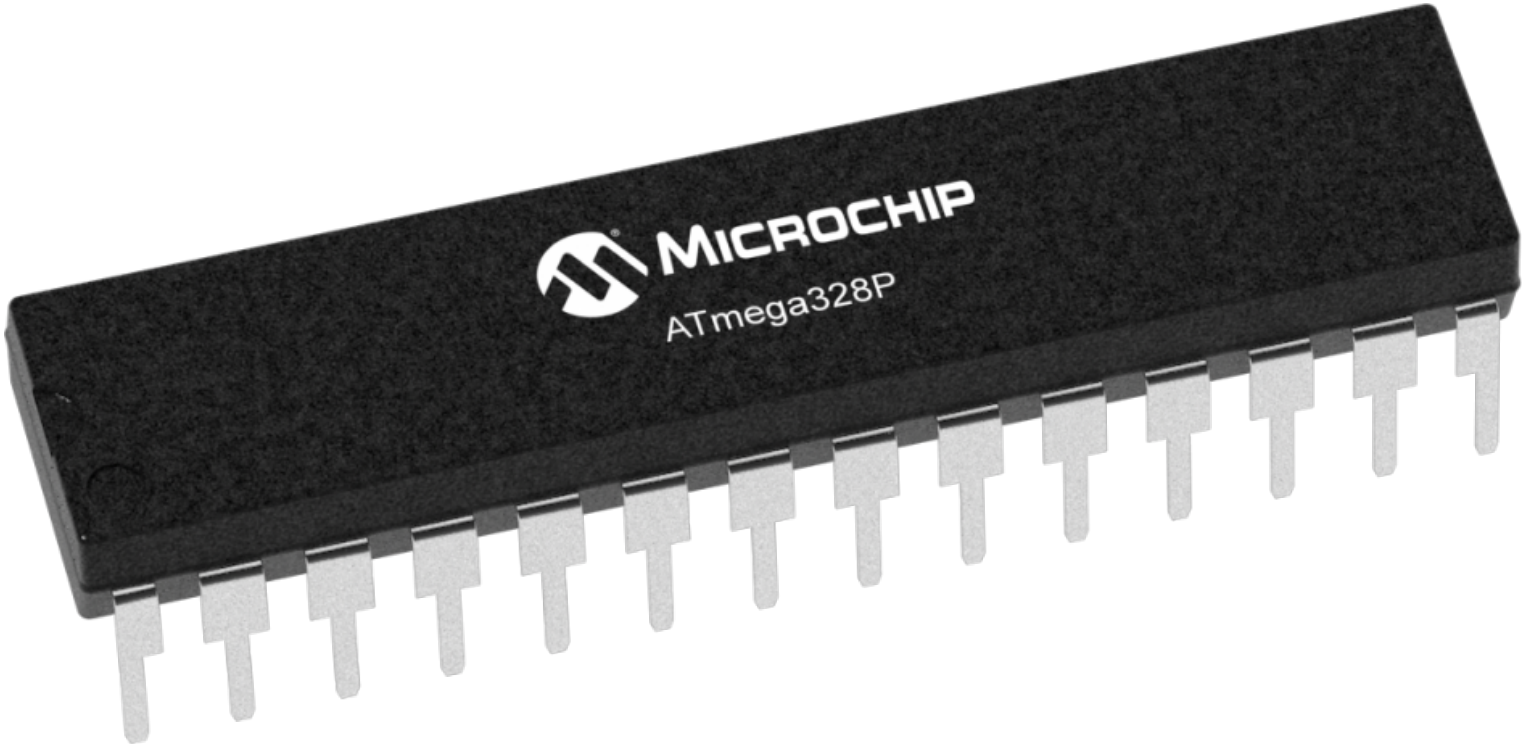
Comparativa con otros fabricantes

	8051	PIC	AVR
Velocidad	Lento	Medio	Rápido
Memoria	Pequeña	Grande	Grande
Arquitectura	CISC	RISC	RISC
ADC	No incluido	Integrado	Integrado
Temporizad.	Integrado	Integrado	Integrado
Canales PWM	No incluido	Integrado	Integrado

Arquitectura de 32 bits: AVR32

- Presentada en 2006
- Diferente de la de 8 bits para competir con ARM
- Instrucciones SIMD y DSP
- RISC pero incompatible con AVR 8 de bits
- Al obtener ATMEL la licencia para desarrollo de ARM Cortex-M y Cortex-A queda relegada a un segundo plano

Atmega328P



Familia

- ATmega48A/PA/88A/PA/168A/PA/328/P

Comparison Between Processors

The ATmega48A/PA/88A/PA/168A/PA/328/P differ only in memory sizes, boot loader support, and interrupt vector sizes. Table 2-1 summarizes the different memory and interrupt vector sizes for the devices.

Table 2-1. Memory Size Summary

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48A	4KBytes	256Bytes	512Bytes	1 instruction word/vector
ATmega48PA	4KBytes	256Bytes	512Bytes	1 instruction word/vector
ATmega88A	8KBytes	512Bytes	1KBytes	1 instruction word/vector
ATmega88PA	8KBytes	512Bytes	1KBytes	1 instruction word/vector
ATmega168A	16KBytes	512Bytes	1KBytes	2 instruction words/vector
ATmega168PA	16KBytes	512Bytes	1KBytes	2 instruction words/vector
ATmega328	32KBytes	1KBytes	2KBytes	2 instruction words/vector
ATmega328P	32KBytes	1KBytes	2KBytes	2 instruction words/vector

Familia

- ATmega48A/PA/88A/PA/168A/PA/328/P

"ATmega88A/PA/168A/PA/328/P support a real Read-While-Write Self-Programming mechanism.

There is a separate Boot Loader Section, and the SPM instruction can only execute from there.

In Atmega48A/48PA there is no Read-While-Write support and no separate Boot Loader Section.

The SPM instruction can execute from the entire Flash."

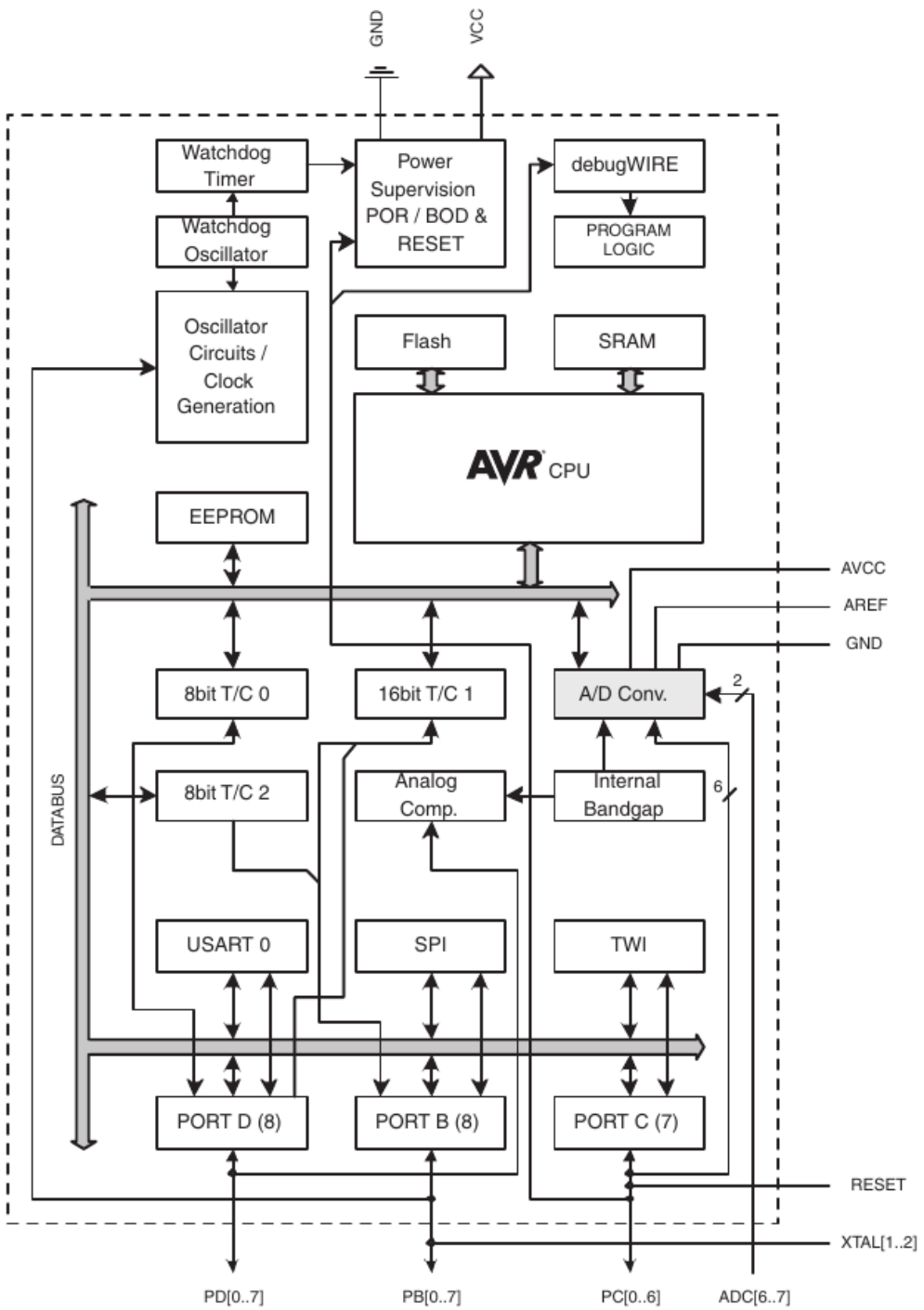
Características ATmega328P

Parameter	Value
CPU type	8-bit AVR
Performance	20 MIPS at 20 MHz
Flash memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Pin count	28 PDIP (32 other)
Maximum operating frequency	20 MHz
Number of touch channels	16
Hardware QTouch Acquisition	No
Maximum I/O pins	23
Interrupt sources (external)	25 (2)
USB Interface	No
USB Speed	–

Data Retention

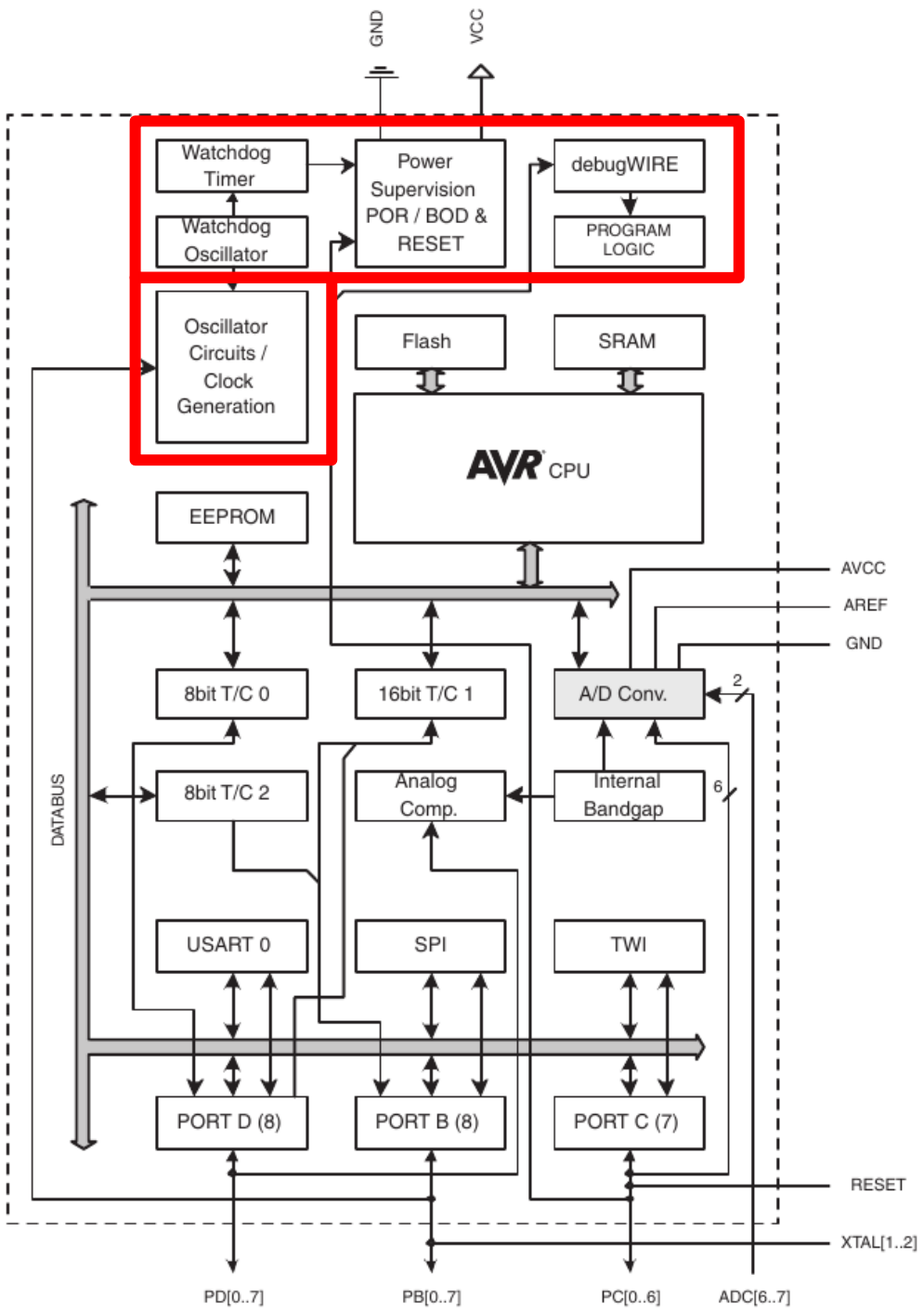
Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

Arquitectura ATmega328P



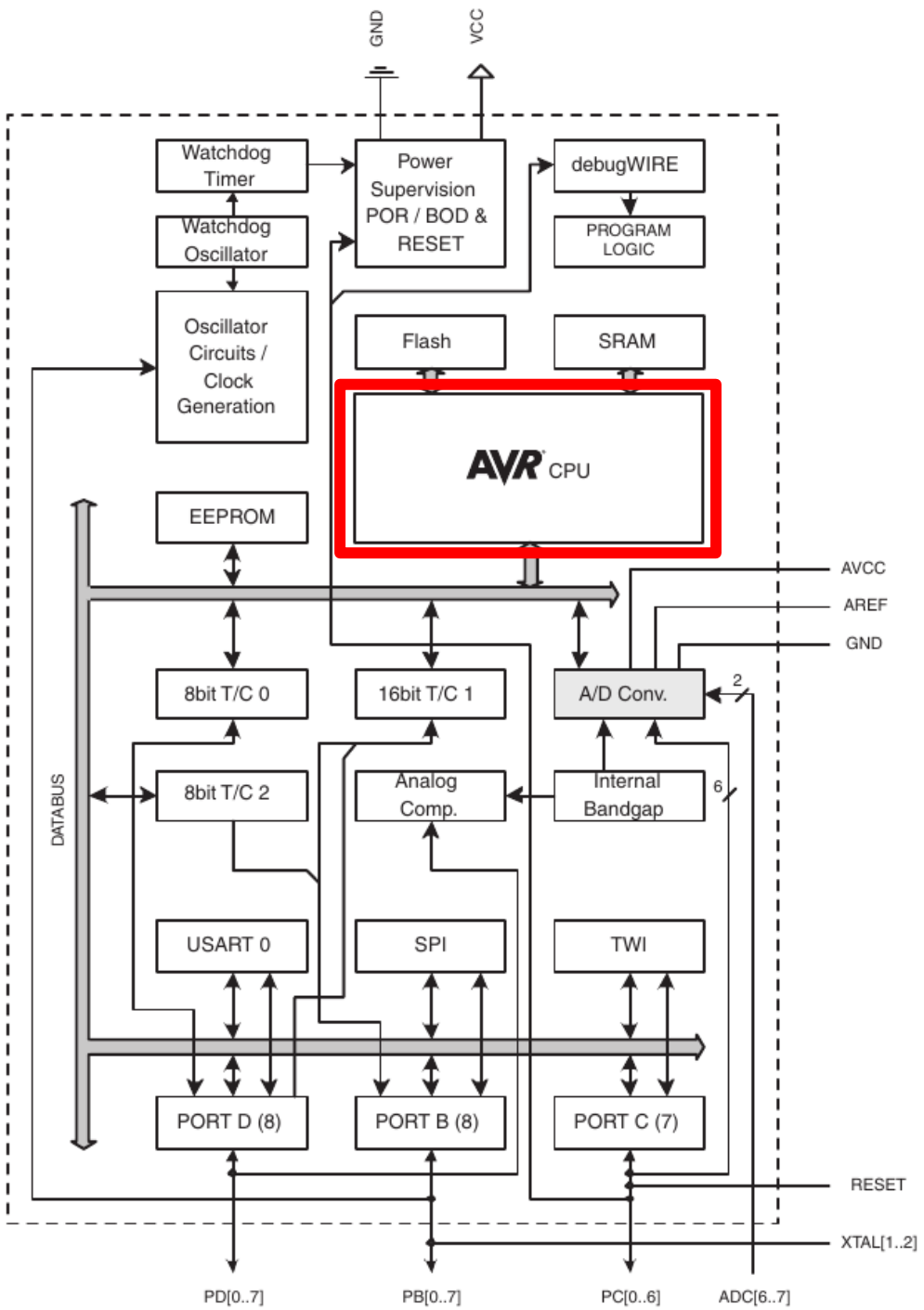
Arquitectura ATmega328P

- Alimentación
- Señales de reloj



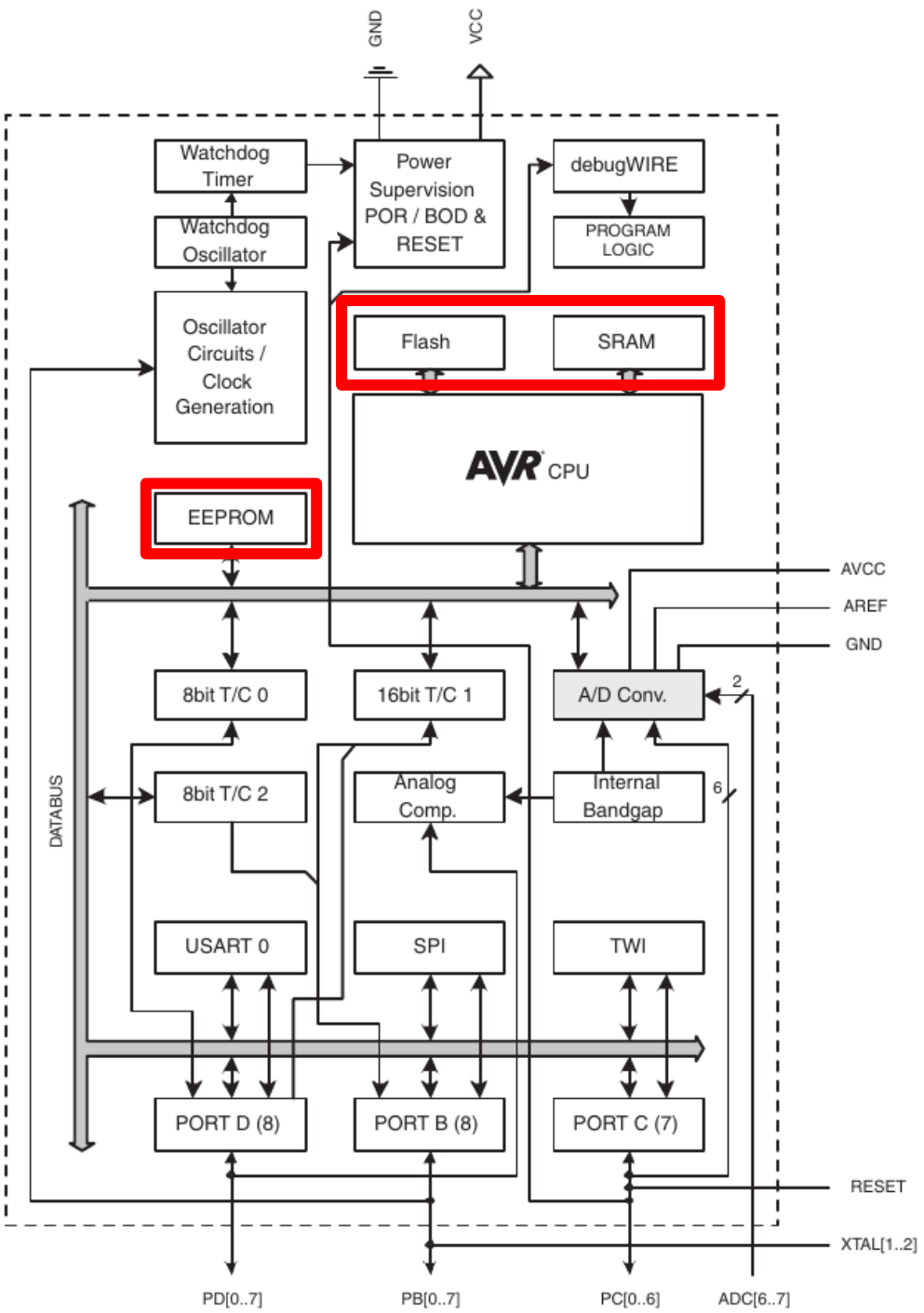
Arquitectura ATmega328P

- Núcleo procesador



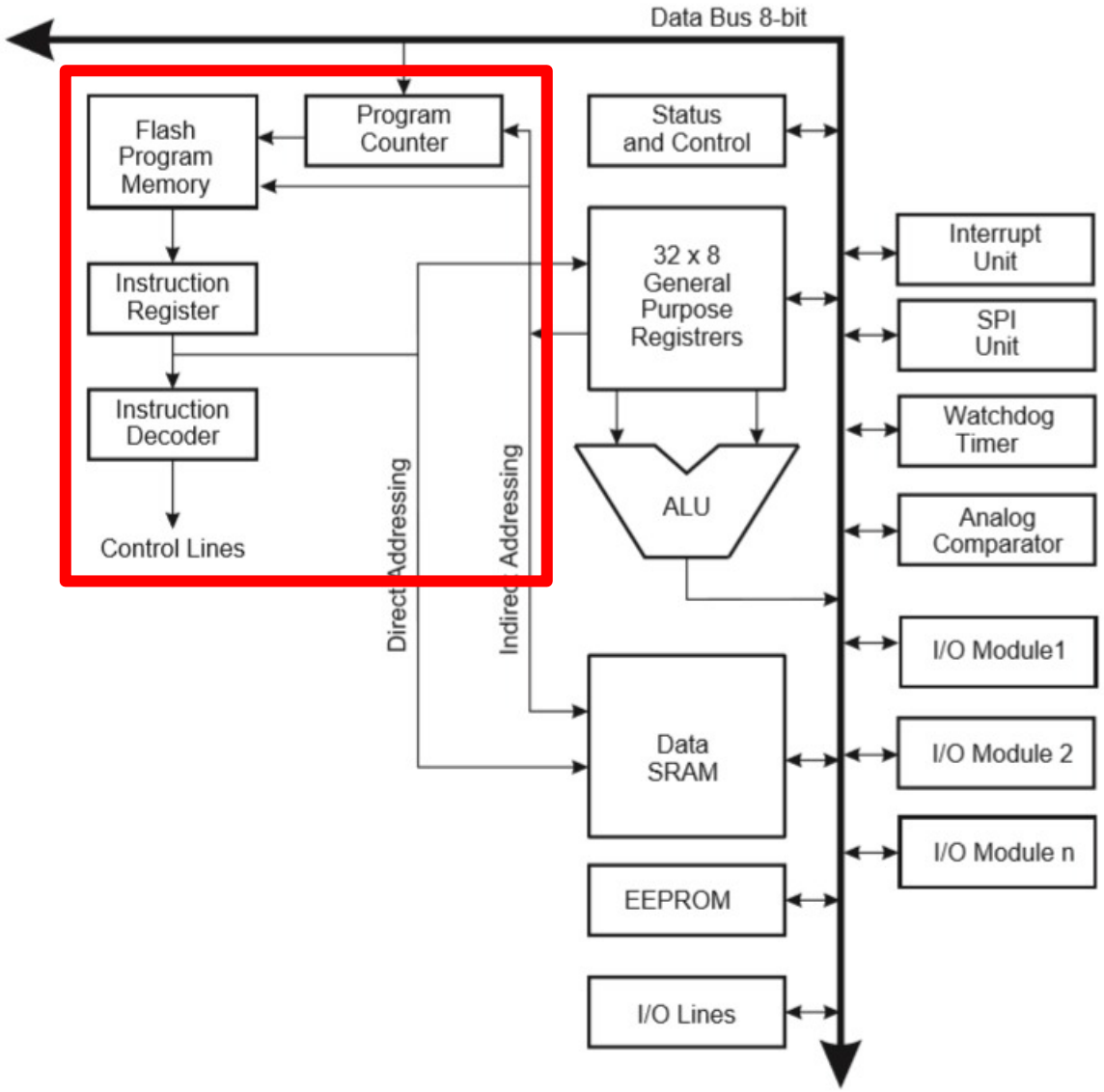
Arquitectura ATmega328P

- Arquitectura Harvard
- Memoria de programa
  Flash: 32KB
- Memoria de datos
  SRAM: 2KB
- EEPROM no volátil
  en el bus de datos: 1KB



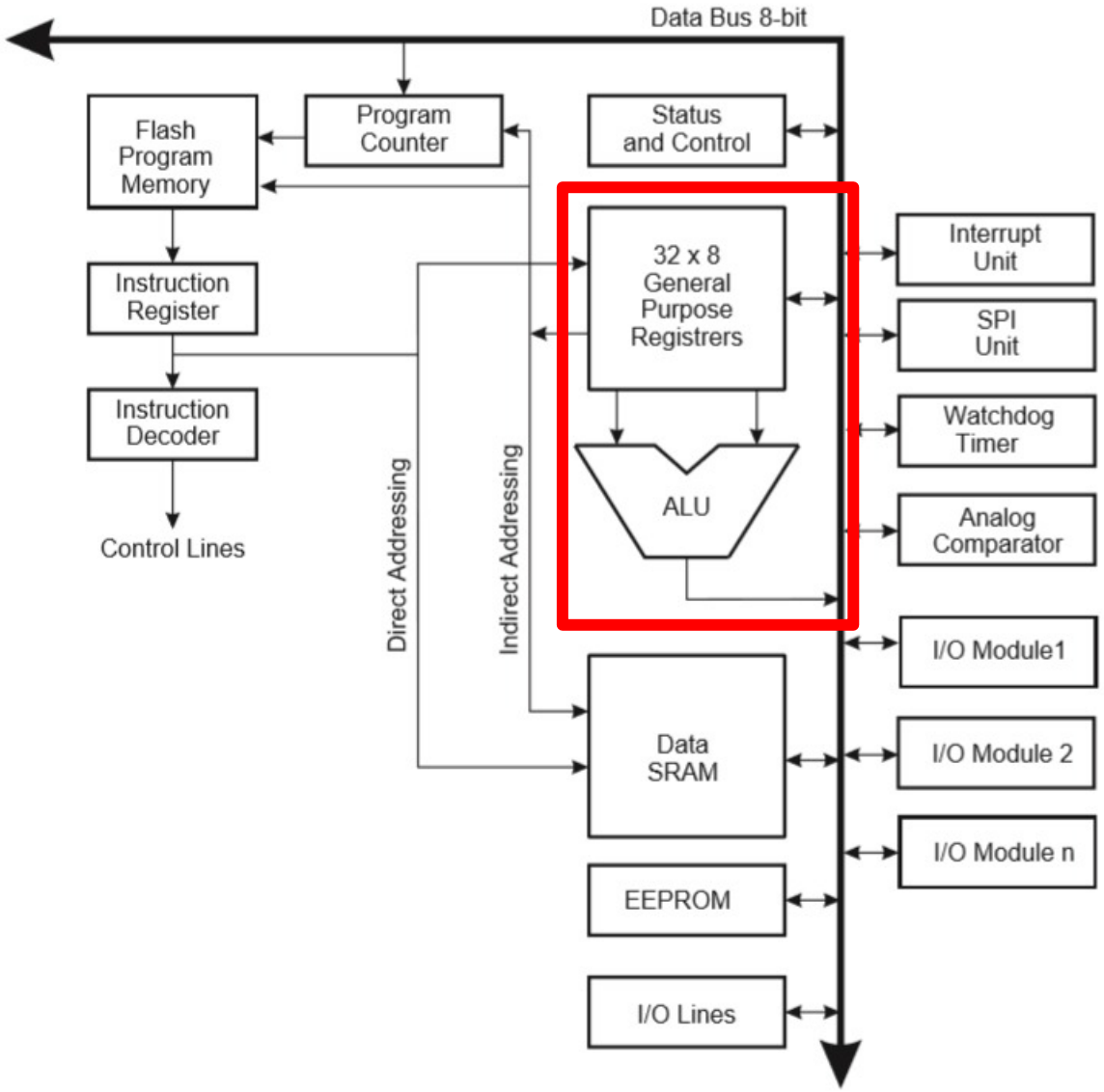
AVR CPU

- Captación y decodificación de instrucción



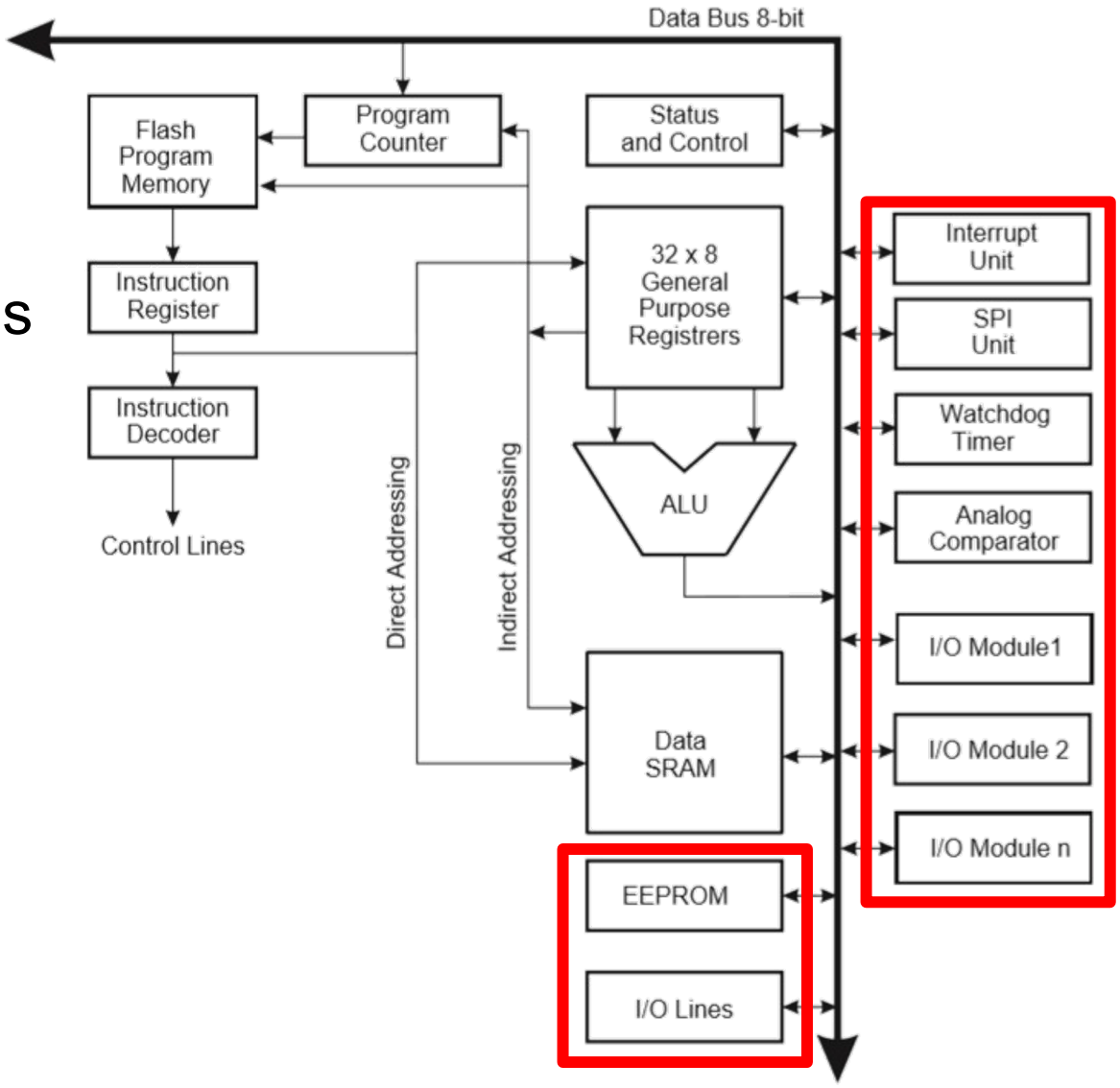
AVR CPU

- Operaciones con la unidad aritmético-lógica (ALU)
- Ambos operandos en registros de propósito general (GPR)



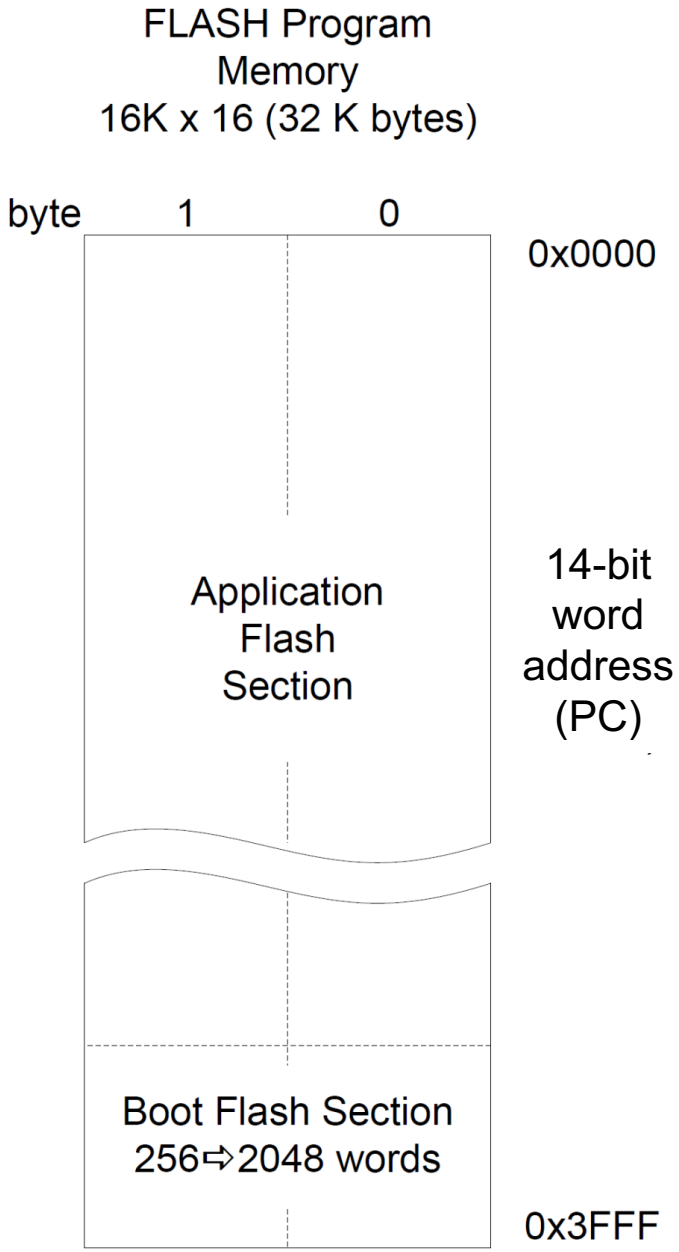
AVR CPU

- Entradas/Salidas
- Funciones periféricas y especiales



Organización de las memorias

- Memoria de programa
- *Boot flash section* protegida por *boot lock bits*
- Puede programarse mediante SPI o en paralelo (FLASH, EEPROM, *memory lock bits* y *fuse bits*)
- Al menos 10.000 ciclos de lectura/escritura



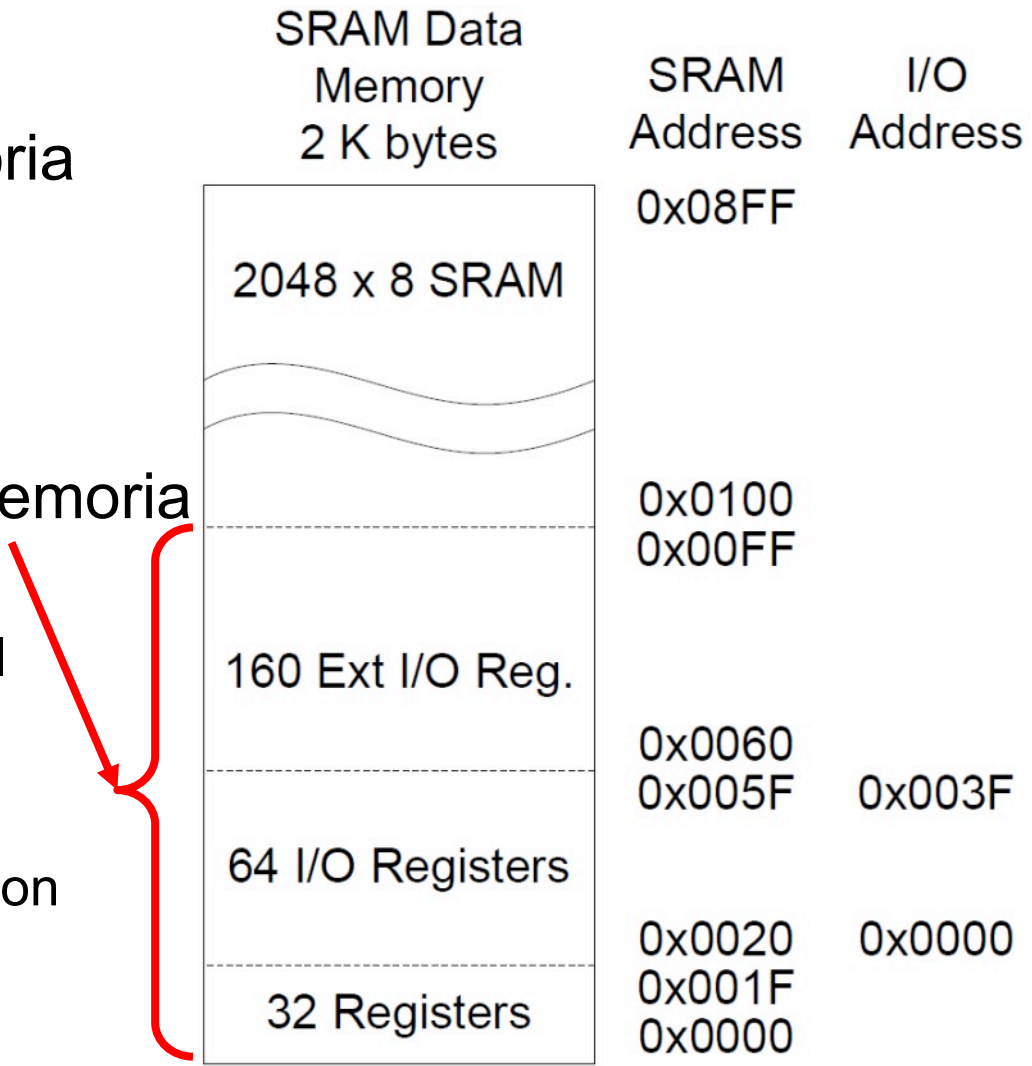
Organización de las memorias

- Direcciones especiales: vectores de interrupciones y de reinicio

Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External pin, power-on reset, brown-out reset and watchdog system reset
2	0x002	INT0	External interrupt request 0
3	0x0004	INT1	External interrupt request 1
4	0x0006	PCINT0	Pin change interrupt request 0
5	0x0008	PCINT1	Pin change interrupt request 1
6	0x000A	PCINT2	Pin change interrupt request 2
7	0x000C	WDT	Watchdog time-out interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 compare match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 compare match B
10	0x0012	TIMER2 OVF	Timer/Counter2 overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 capture event
12	0x0016	TIMER1 COMPA	Timer/Counter1 compare match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 compare match B
14	0x001A	TIMER1 OVF	Timer/Counter1 overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 compare match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 compare match B
17	0x0020	TIMER0 OVF	Timer/Counter0 overflow
18	0x0022	SPI, STC	SPI serial transfer complete
19	0x0024	USART, RX	USART Rx complete
20	0x0026	USART, UDRE	USART, data register empty
21	0x0028	USART, TX	USART, Tx complete
22	0x002A	ADC	ADC conversion complete
23	0x002C	EE READY	EEPROM ready
24	0x002E	ANALOG COMP	Analog comparator
25	0x0030	TWI	2-wire serial interface
26	0x0032	SPM READY	Store program memory ready

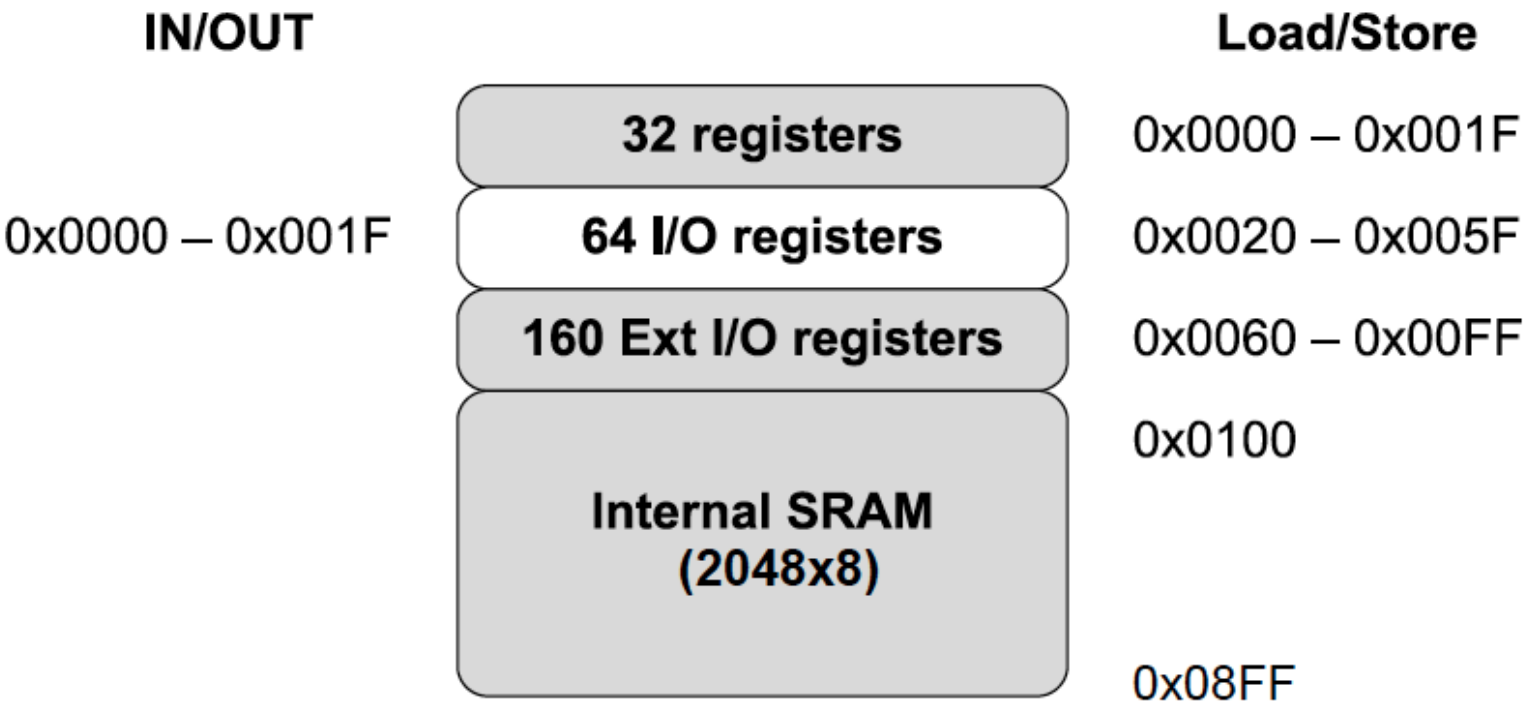
Organización de las memorias

- Memoria RAM (datos)
- 2048 posiciones de memoria
  - Incluye espacio de pila
- Recursos mapeados en el espacio de direcciones de memoria
  - 32 reg. de propósito general
  - 64 + 160 registros de E/S y E/S extendida (accesibles con diferentes instrucciones)



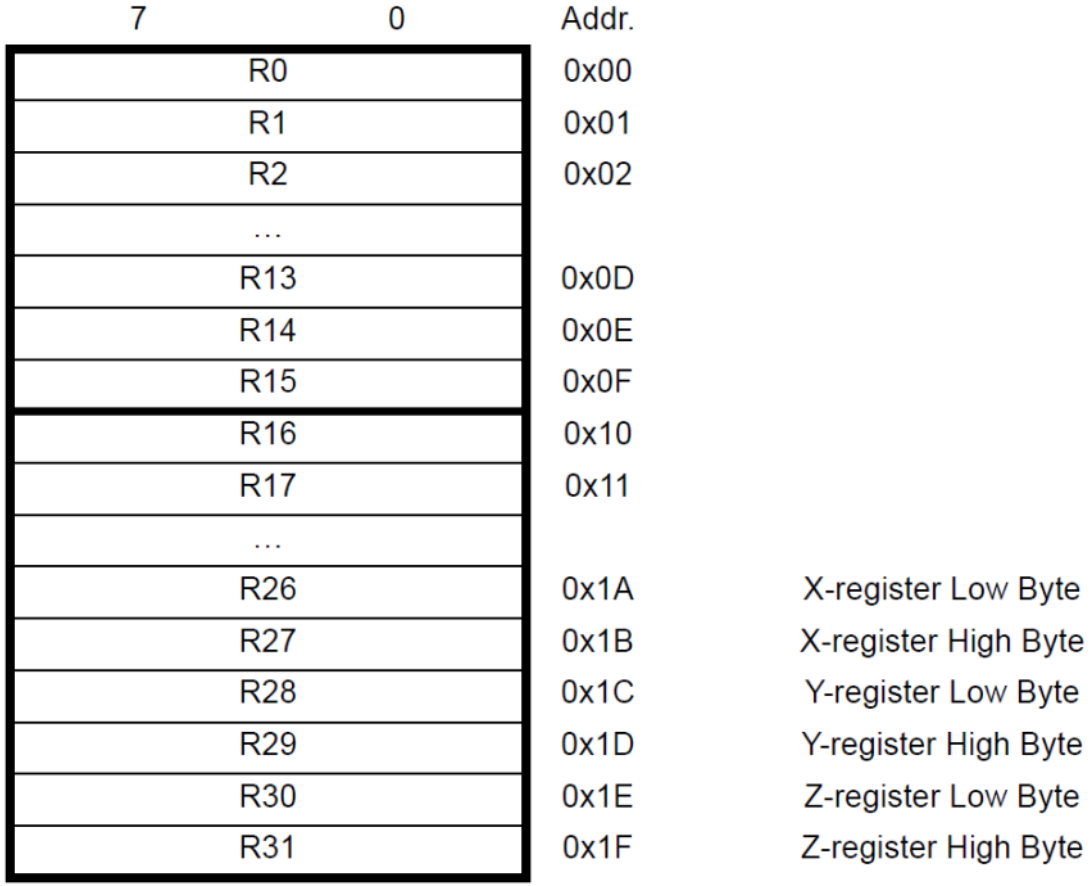
Espacio de entrada/salida

- Registros de E/S accesibles desde el espacio de entrada/salida (instrucciones IN/OUT) y desde el espacio de memoria (instrucciones LD/ST)



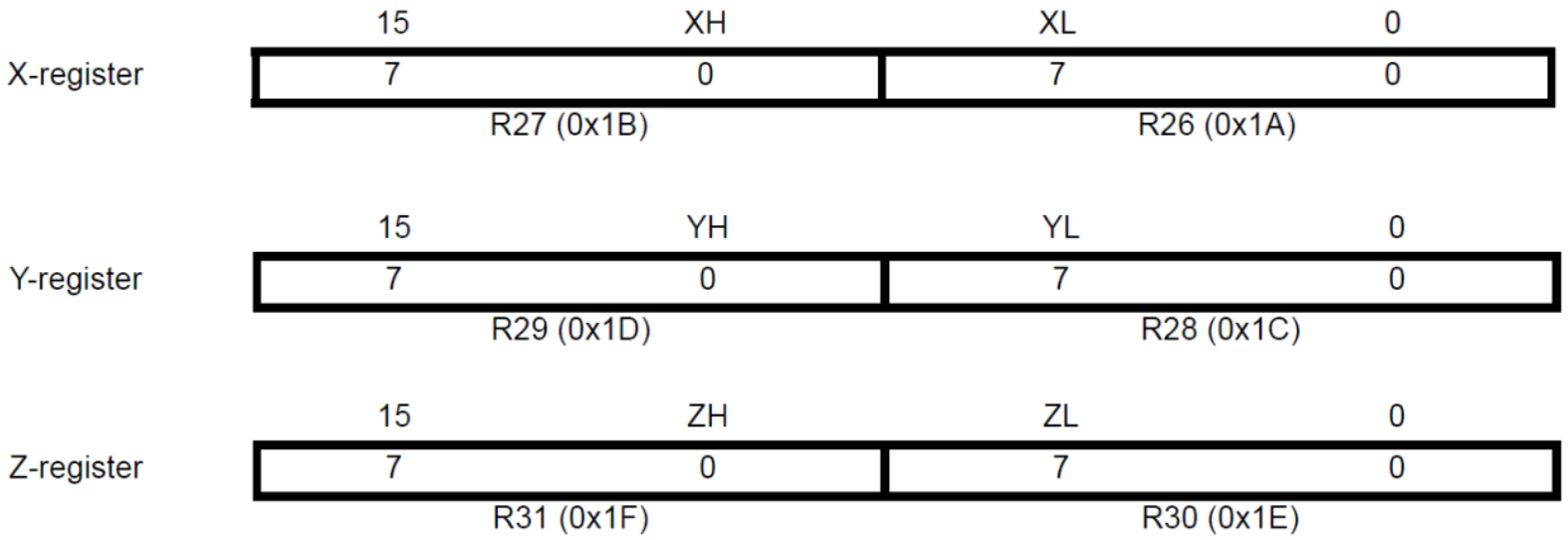
Registros de propósito general

- Optimizados para ser Usados con datos de 8 y 16 bits
- Registros X, Y, Z para direccionamiento indirecto



Registros de propósito general

- Registros X, Y, Z



In the different addressing modes these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the instruction set reference for details).

INFO

I/O and Ext. I/O registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x22 (0x42)	EEARH	(EEPROM address register high byte)							
0x21 (0x41)	EEARL	EEPROM address register low byte							
0x20 (0x40)	EEDR	EEPROM data register							
0x1F (0x3F)	EECR	–	–	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE
0x1E (0x3E)	GPIOR0	General purpose I/O register 0							
0x1D (0x3D)	EIMSK	–	–	–	–	–	–	INT1	INT0
0x1C (0x3C)	EIFR	–	–	–	–	–	–	INTF1	INTF0
0x1B (0x3B)	PCIFR	–	–	–	–	–	PCIF2	PCIF1	PCIF0
0x1A (0x3A)	Reserved	–	–	–	–	–	–	–	–
0x19 (0x39)	Reserved	–	–	–	–	–	–	–	–
0x18 (0x38)	Reserved	–	–	–	–	–	–	–	–
0x17 (0x37)	TIFR2	–	–	–	–	–	OCF2B	OCF2A	TOV2
0x16 (0x36)	TIFR1	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1
0x15 (0x35)	TIFR0	–	–	–	–	–	OCF0B	OCF0A	TOV0
0x14 (0x34)	Reserved	–	–	–	–	–	–	–	–
0x13 (0x33)	Reserved	–	–	–	–	–	–	–	–
0x12 (0x32)	Reserved	–	–	–	–	–	–	–	–
0x11 (0x31)	Reserved	–	–	–	–	–	–	–	–
0x10 (0x30)	Reserved	–	–	–	–	–	–	–	–
0x0F (0x2F)	Reserved	–	–	–	–	–	–	–	–
0x0E (0x2E)	Reserved	–	–	–	–	–	–	–	–
0x0D (0x2D)	Reserved	–	–	–	–	–	–	–	–
0x0C (0x2C)	Reserved	–	–	–	–	–	–	–	–
0x0B (0x2B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
0x0A (0x2A)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
0x09 (0x29)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
0x08 (0x28)	PORTC	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0
0x07 (0x27)	DDRC	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
0x06 (0x26)	PINC	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
0x05 (0x25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
0x04 (0x24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
0x03 (0x23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
0x02 (0x22)	Reserved	–	–	–	–	–	–	–	–
0x01 (0x21)	Reserved	–	–	–	–	–	–	–	–
0x0 (0x20)	Reserved	–	–	–	–	–	–	–	–

INFO

I/O and Ext. I/O registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(0x68)	PCICR	–	–	–	–	–	PCIE2	PCIE1	PCIE0
(0x67)	Reserved	–	–	–	–	–	–	–	–
(0x66)	OSCCAL	Oscillator calibration register							
(0x65)	Reserved	–	–	–	–	–	–	–	–
(0x64)	PRR	PRTWI	PRTIM2	PRTIM0	–	PRTIM1	PRSPI	PRUSAR0	PRADC
(0x63)	Reserved	–	–	–	–	–	–	–	–
(0x62)	Reserved	–	–	–	–	–	–	–	–
(0x61)	CLKPR	CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS0
(0x60)	WDTCSR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0
0x3F (0x5F)	SREG	I	T	H	S	V	N	Z	C
0x3E (0x5E)	SPH	–	–	–	–	–	(SP10)	SP9	SP8
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
0x3C (0x5C)	Reserved	–	–	–	–	–	–	–	–
0x3B (0x5B)	Reserved	–	–	–	–	–	–	–	–
0x3A (0x5A)	Reserved	–	–	–	–	–	–	–	–
0x39 (0x59)	Reserved	–	–	–	–	–	–	–	–
0x38 (0x58)	Reserved	–	–	–	–	–	–	–	–
0x37 (0x57)	SPMCSR	SPMIE	(RWWSB)	–	(RWWSRE)	BLBSET	PGWRT	PGERS	SELFPRGN
0x36 (0x56)	Reserved	–	–	–	–	–	–	–	–
0x35 (0x55)	MCUCR	–	BODS	BODSE	PUD	–	–	IVSEL	IVCE
0x34 (0x54)	MCUSR	–	–	–	–	WDRF	BORF	EXTRF	PORF
0x33 (0x53)	SMCR	–	–	–	–	SM2	SM1	SM0	SE
0x32 (0x52)	Reserved	–	–	–	–	–	–	–	–
0x31 (0x51)	Reserved	–	–	–	–	–	–	–	–
0x30 (0x50)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0
0x2F (0x4F)	Reserved	–	–	–	–	–	–	–	–
0x2E (0x4E)	SPDR	SPI data register							
0x2D (0x4D)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X
0x2C (0x4C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
0x2B (0x4B)	GPIOR2	General purpose I/O register 2							
0x2A (0x4A)	GPIOR1	General purpose I/O register 1							
0x29 (0x49)	Reserved	–	–	–	–	–	–	–	–
0x28 (0x48)	OCR0B	Timer/Counter0 output compare register B							
0x27 (0x47)	OCR0A	Timer/Counter0 output compare register A							
0x26 (0x46)	TCNT0	Timer/Counter0 (8-bit)							
0x25 (0x45)	TCCR0B	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00
0x24 (0x44)	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00
0x23 (0x43)	GTCCR	TSM	–	–	–	–	–	PSRASY	PSRSYNC

Y más…

INFO

Detalle I/O register

SREG – AVR Status Register

The AVR status register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I: Global Interrupt Enable**

The global interrupt enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the global interrupt enable register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- **Bit 6 – T: Bit Copy Storage**

The bit copy instructions BLD (bit LoaD) and BST (Bit STore) use the T-bit as source or destination for the operated bit. A bit from a register in the register file can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the register file by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The half carry flag H indicates a half carry in some arithmetic operations. Half carry Is useful in BCD arithmetic. See Section "" on page 281 for detailed information.

- **Bit 4 – S: Sign Bit, S = N ⊕ V**

The S-bit is always an exclusive or between the negative flag N and the two's complement overflow flag V. See Section "" on page 281 for detailed information.

- **Bit 3 – V: Two's Complement Overflow Flag**

The two's complement overflow flag V supports two's complement arithmetics. See Section "" on page 281 for detailed information.

- **Bit 2 – N: Negative Flag**

The negative flag N indicates a negative result in an arithmetic or logic operation. See Section "" on page 281 for detailed information.

- **Bit 1 – Z: Zero Flag**

The zero flag Z indicates a zero result in an arithmetic or logic operation. See Section "" on page 281 for detailed information.

- **Bit 0 – C: Carry Flag**

The carry flag C indicates a carry in an arithmetic or logic operation. See Section "" on page 281 for detailed information.

INFO

Detalle I/O register

PORTB – The Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

DDRB – The Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PINB – The Port B Input Pins Address

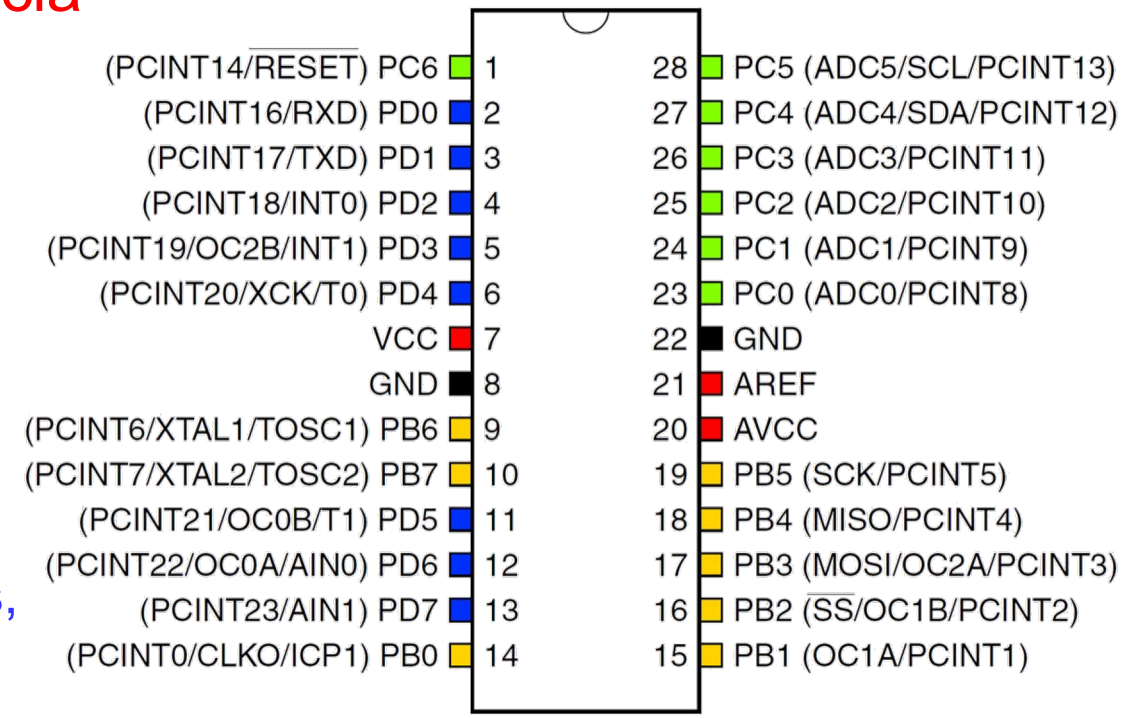
Bit	7	6	5	4	3	2	1	0	
0x03 (0x23)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Table 13-3. Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB7	XTAL2 (chip clock oscillator pin 2) TOSC2 (timer oscillator pin 2) PCINT7 (pin change interrupt 7)
PB6	XTAL1 (chip clock oscillator pin 1 or external clock input) TOSC1 (timer oscillator pin 1) PCINT6 (pin change interrupt 6)
PB5	SCK (SPI bus master clock input) PCINT5 (pin change interrupt 5)
PB4	MISO (SPI bus master input/slave output) PCINT4 (pin change interrupt 4)
PB3	MOSI (SPI bus master output/slave input) OC2A (Timer/Counter2 output compare match A output) PCINT3 (pin change interrupt 3)
PB2	$\overline{SS}$ (SPI bus master slave select) OC1B (Timer/Counter1 output compare match B output) PCINT2 (pin change interrupt 2)
PB1	OC1A (Timer/Counter1 output compare match A output) PCINT1 (pin change interrupt 1)
PB0	ICP1 (Timer/Counter1 input capture input) CLKO (divided system clock output) PCINT0 (pin change interrupt 0)

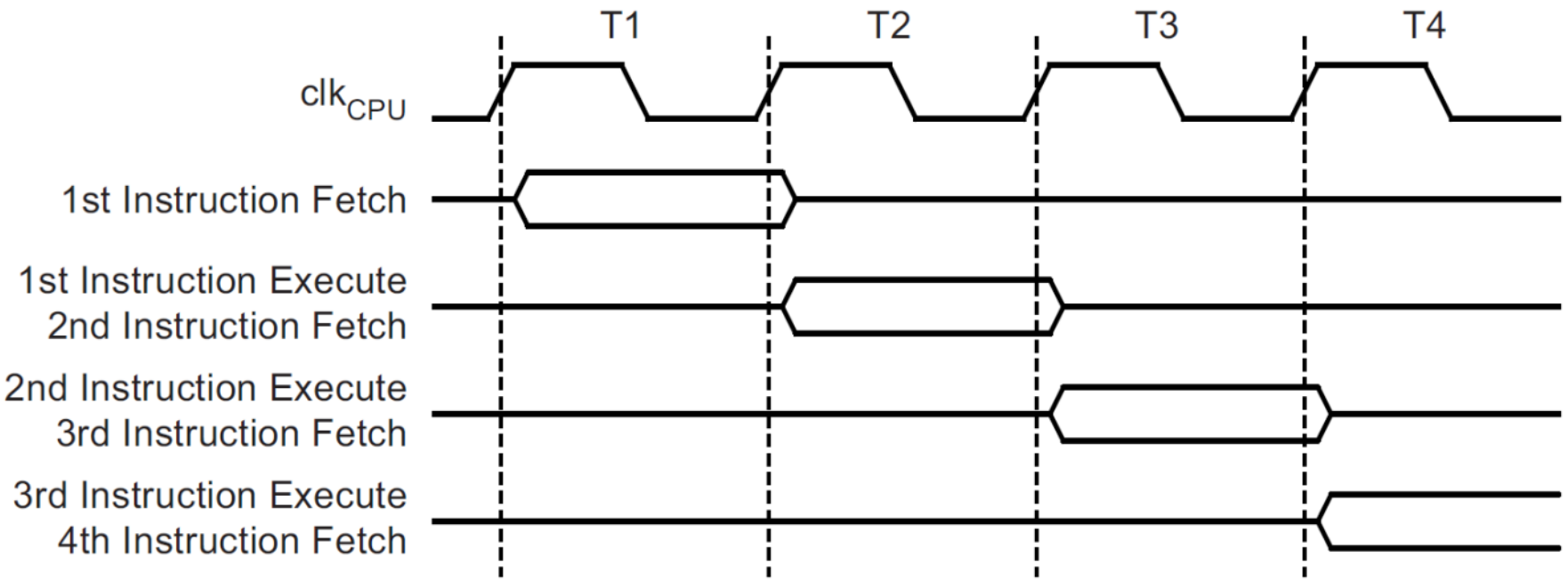
Distribución y funciones de los pines

- Alimentación y referencia
- Masa
- Puerto B
  - Osciladores, SPI, …
- Puerto C
  - ADC, I2C, Reset
- Puerto D
  - Interrupciones externas, UART, …



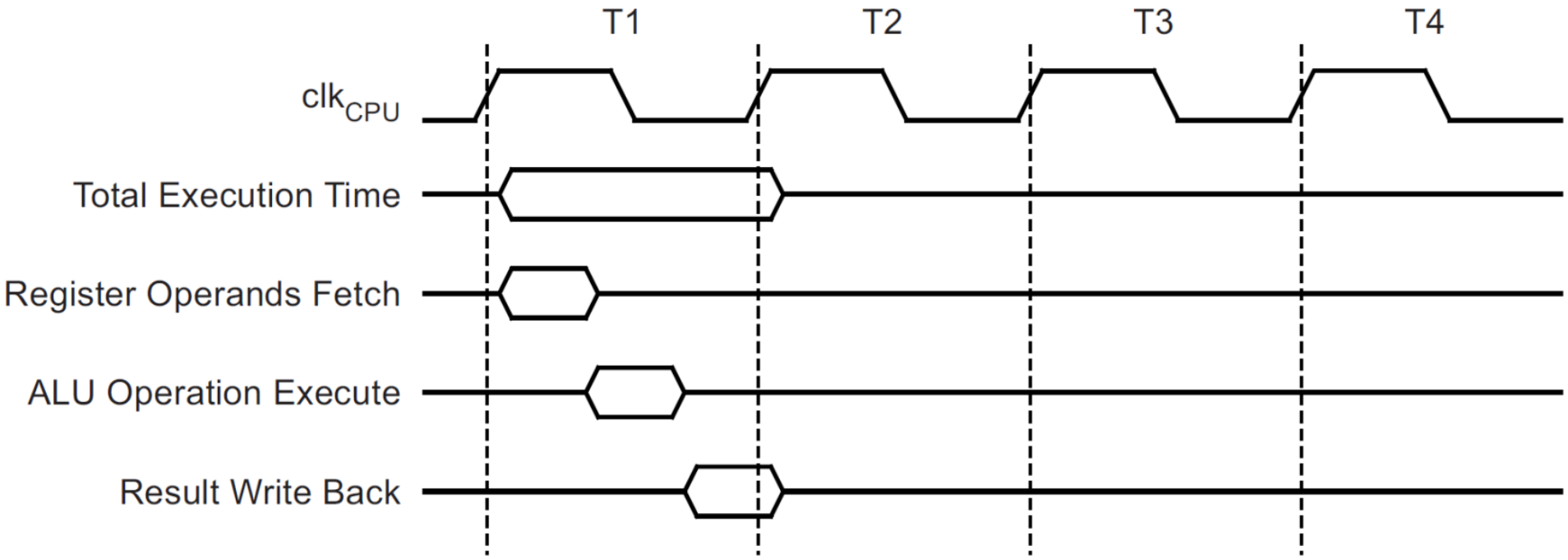
Segmentación de cauce

- 2 etapas



Ciclo de instrucción

- CPI = 1 → MHz ≡ MIPS



INFO

Repertorio de instrucciones

Mnemonics	Operands	Description	Operation	Flags	#Clocks
Arithmetic and Logic Instructions					
ADD	Rd, Rr	Add two registers	Rd ← Rd + Rr	Z,C,N,VH	1
ADC	Rd, Rr	Add with carry two registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rdl, K	Add immediate to word	Rdh: Rdl ← Rdh: Rdl + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two registers	Rd ← Rd – Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract constant from register	Rd ← Rd – K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with carry two registers	Rd ← Rd – Rr – C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with carry constant from reg.	Rd ← Rd – K – C	Z,C,N,V,H	1
SBIW	Rdl, K	Subtract immediate from word	Rdh: Rdl ← Rdh: Rdl – K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND registers	Rd ← Rd × Rr	Z,N,V	1
ANDI	Rd, K	Logical AND register and constant	Rd ← Rd × K	Z,N,V	1
OR	Rd, Rr	Logical OR registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR register and constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's complement	Rd ← 0xFF – Rd	Z,C,N,V	1
NEG	Rd	Two's complement	Rd ← 0x00 – Rd	Z,C,N,V,H	1
SBR	Rd, K	Set bit(s) in register	Rd ← Rd v K	Z,N,V	1
CBR	Rd, K	Clear bit(s) in register	Rd ← Rd × (0xFF – K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd – 1	Z,N,V	1
TST	Rd	Test for zero or minus	Rd ← Rd × Rd	Z,N,V	1
CLR	Rd	Clear register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set register	Rd ← 0xFF	None	1
MUL	Rd, Rr	Multiply unsigned	R1:R0 ← Rd × Rr	Z,C	2
MULS	Rd, Rr	Multiply signed	R1:R0 ← Rd × Rr	Z,C	2
MULSU	Rd, Rr	Multiply signed with unsigned	R1:R0 ← Rd × Rr	Z,C	2
FMUL	Rd, Rr	Fractional multiply unsigned	R1:R0 ← (Rd × Rr) << 1	Z,C	2
FMULS	Rd, Rr	Fractional multiply signed	R1:R0 ← (Rd × Rr) << 1	Z,C	2
FMULSU	Rd, Rr	Fractional multiply signed with unsigned	R1:R0 ← (Rd × Rr) << 1	Z,C	2

INFO

Repertorio de instrucciones

Branch Instructions					
RJMP	k	Relative jump	PC ← PC + k + 1	None	2
IJMP		Indirect jump to (Z)	PC ← Z	None	2
JMP	k	Direct jump	PC ← k	None	3
RCALL	k	Relative subroutine call	PC ← PC + k + 1	None	3
ICALL		Indirect call to (Z)	PC ← Z	None	3
CALL	k	Direct subroutine call	PC ← k	None	4
RET		Subroutine return	PC ← STACK	None	4
RETI		Interrupt return	PC ← STACK	I	4
CPSE	Rd, Rr	Compare, skip if equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1/2/3
CP	Rd, Rr	Compare	Rd – Rr	Z,N,V,C,H	1
CPC	Rd, Rr	Compare with carry	Rd – Rr – C	Z, N,V,C,H	1
CPI	Rd, K	Compare register with immediate	Rd – K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if bit in register cleared	if (Rr (b) = 0) PC ← PC + 2 or 3	None	1/2/3
SBRS	Rr, b	Skip if bit in register is set	if (Rr(b)=1) PC ← PC + 2 or 3	None	1/2/3
SBIC	P, b	Skip if bit in I/O register cleared	if (P(b)=0) PC ← PC + 2 or 3	None	1/2/3
SBIS	P, b	Skip if bit in I/O register is set	if (P(b)=1) PC ← PC + 2 or 3	None	1/2/3
BRBS	s, k	Branch if status flag set	if (SREG(s) = 1) then PC←PC + k + 1	None	1/2
BRBC	s, k	Branch if status flag cleared	if (SREG(s) = 0) then PC←PC + k + 1	None	1/2
BREQ	k	Branch if equal	if (Z = 1) then PC ← PC + k + 1	None	1/2
BRNE	k	Branch if not equal	if (Z = 0) then PC ← PC + k + 1	None	1/2
BRCS	k	Branch if carry set	if (C = 1) then PC ← PC + k + 1	None	1/2
BRCC	k	Branch if carry cleared	if (C = 0) then PC ← PC + k + 1	None	1/2
BRSH	k	Branch if same or higher	if (C = 0) then PC ← PC + k + 1	None	1/2
BRLO	k	Branch if lower	if (C = 1) then PC ← PC + k + 1	None	1/2
BRMI	k	Branch if minus	if (N = 1) then PC ← PC + k + 1	None	1/2
BRPL	k	Branch if plus	if (N = 0) then PC ← PC + k + 1	None	1/2
BRGE	k	Branch if greater or equal, signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1/2
BRLT	k	Branch if less than zero, signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1/2
BRHS	k	Branch if half carry flag set	if (H = 1) then PC ← PC + k + 1	None	1/2
BRHC	k	Branch if half carry flag cleared	if (H = 0) then PC ← PC + k + 1	None	1/2
BRTS	k	Branch if T flag set	if (T = 1) then PC ← PC + k + 1	None	1/2
BRTC	k	Branch if T flag cleared	if (T = 0) then PC ← PC + k + 1	None	1/2
BRVS	k	Branch if overflow flag is set	if (V = 1) then PC ← PC + k + 1	None	1/2
BRVC	k	Branch if overflow flag is cleared	if (V = 0) then PC ← PC + k + 1	None	1/2
BRIE	k	Branch if interrupt enabled	if (I = 1) then PC ← PC + k + 1	None	1/2
BRID	k	Branch if interrupt disabled	if (I = 0) then PC ← PC + k + 1	None	1/2

INFO

Repertorio de instrucciones

Bit and Bit-Test Instructions					
SBI	P, b	Set bit in I/O register	I/O (P, b) ← 1	None	2
CBI	P, b	Clear bit in I/O register	I/O (P, b) ← 0	None	2
LSL	Rd	Logical shift left	Rd(n+1) ← Rd (n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical shift right	Rd (n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate left through carry	Rd(0) ← C,Rd(n+1) ← Rd (n), C ← Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate right through carry	Rd(7) ← C, Rd (n) ← Rd(n+1),C ← Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic shift right	Rd (n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap nibbles	Rd(3..0) ← Rd(7..4),Rd(7..4) ← Rd(3..0)	None	1
BSET	s	Flag set	SREG (s) ← 1	SREG (s)	1
BCLR	s	Flag clear	SREG (s) ← 0	SREG (s)	1
BST	Rr, b	Bit store from register to T	T ← Rr (b)	T	1
BLD	Rd, b	Bit load from T to register	Rd (b) ← T	None	1
SEC		Set carry	C ← 1	C	1
CLC		Clear carry	C ← 0	C	1
SEN		Set negative flag	N ← 1	N	1
CLN		Clear negative flag	N ← 0	N	1
SEZ		Set zero flag	Z ← 1	Z	1
CLZ		Clear zero flag	Z ← 0	Z	1
SEI		Global interrupt enable	I ← 1	I	1
CLI		Global interrupt disable	I ← 0	I	1
SES		Set signed test flag	S ← 1	S	1
CLS		Clear signed test flag	S ← 0	S	1
SEV		Set twos complement overflow.	V ← 1	V	1
CLV		Clear twos complement overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set half carry flag in SREG	H ← 1	H	1
CLH		Clear half carry flag in SREG	H ← 0	H	1

INFO

Repertorio de instrucciones

Data Transfer Instructions					
MOV	Rd, Rr	Move between registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy register word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load immediate	Rd ← K	None	1
LD	Rd, X	Load indirect	Rd ← (X)	None	2
LD	Rd, X+	Load indirect and post-inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, – X	Load indirect and pre-dec.	X ← X – 1, Rd ← (X)	None	2
LD	Rd, Y	Load indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load indirect and post-inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, – Y	Load indirect and pre-dec.	Y ← Y – 1, Rd ← (Y)	None	2
LDD	Rd, Y+ q	Load indirect with displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load indirect and post-inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, –Z	Load indirect and pre-dec.	Z ← Z – 1, Rd ← (Z)	None	2
LDD	Rd, Z+ q	Load indirect with displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store indirect	(X) ← Rr	None	2
ST	X+, Rr	Store indirect and post-inc.	(X) ← Rr, X ← X + 1	None	2
ST	– X, Rr	Store indirect and pre-dec.	X ← X – 1, (X) ← Rr	None	2
ST	Y, Rr	Store indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store indirect and post-inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	– Y, Rr	Store indirect and pre-dec.	Y ← Y – 1, (Y) ← Rr	None	2
STD	Y+ q, Rr	Store indirect with displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store indirect	(Z) ← Rr	None	2
ST	Z +, Rr	Store indirect and post-inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	–Z, Rr	Store indirect and pre-dec.	Z ← Z – 1, (Z) ← Rr	None	2
STD	Z + q, Rr	Store indirect with displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store direct to SRAM	(k) ← Rr	None	2
LPM		Load program memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load program memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load program memory and post-inc	Rd ← (Z), Z ← Z+1	None	3
SPM		Store program memory	(Z) ← R1:R0	None	–
IN	Rd, P	In port	Rd ← P	None	1
OUT	P, Rr	Out port	P ← Rr	None	1
PUSH	Rr	Push register on stack	STACK ← Rr	None	2
POP	Rd	Pop register from stack	Rd ← STACK	None	2

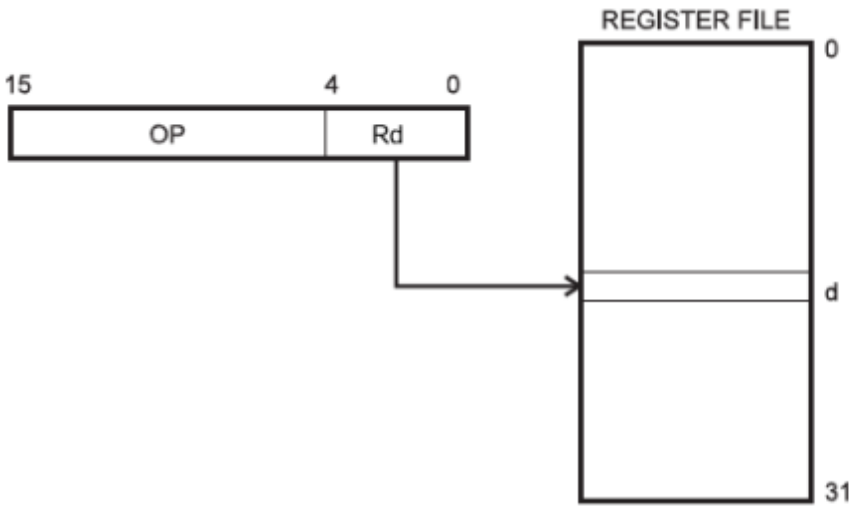
INFO

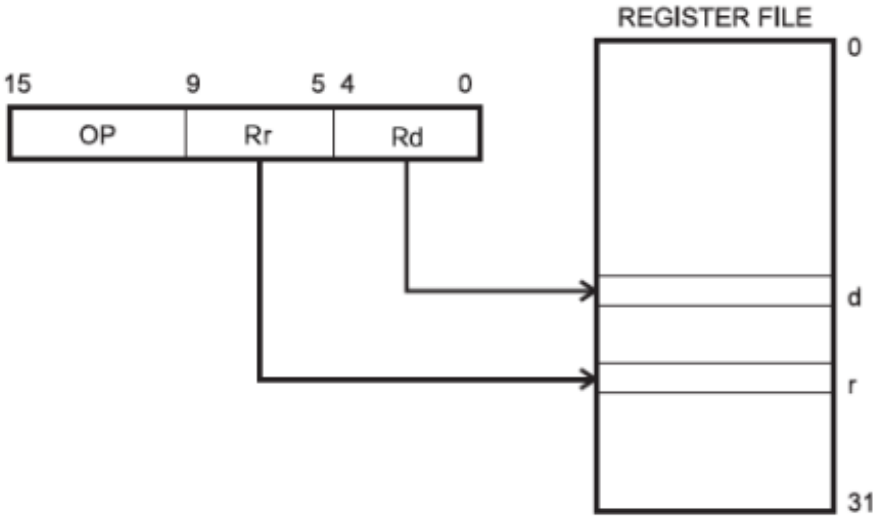
Repertorio de instrucciones

MCU Control Instructions					
NOP		No operation		None	1
SLEEP		Sleep	(see specific descr. for sleep function)	None	1
WDR		Watchdog reset	(see specific descr. for WDR/timer)	None	1
BREAK		Break	For on-chip debug only	None	N/A

Modos de direccionamiento

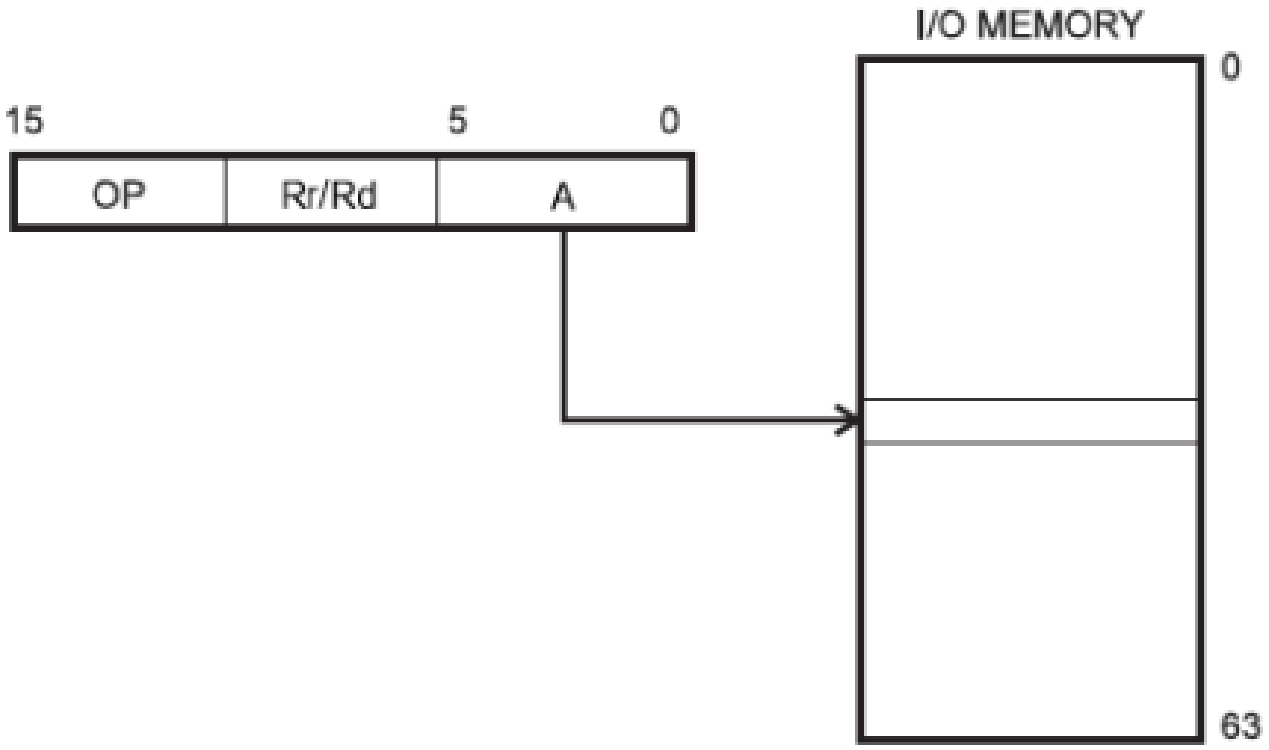
- Directo a registro (1 ó 2)





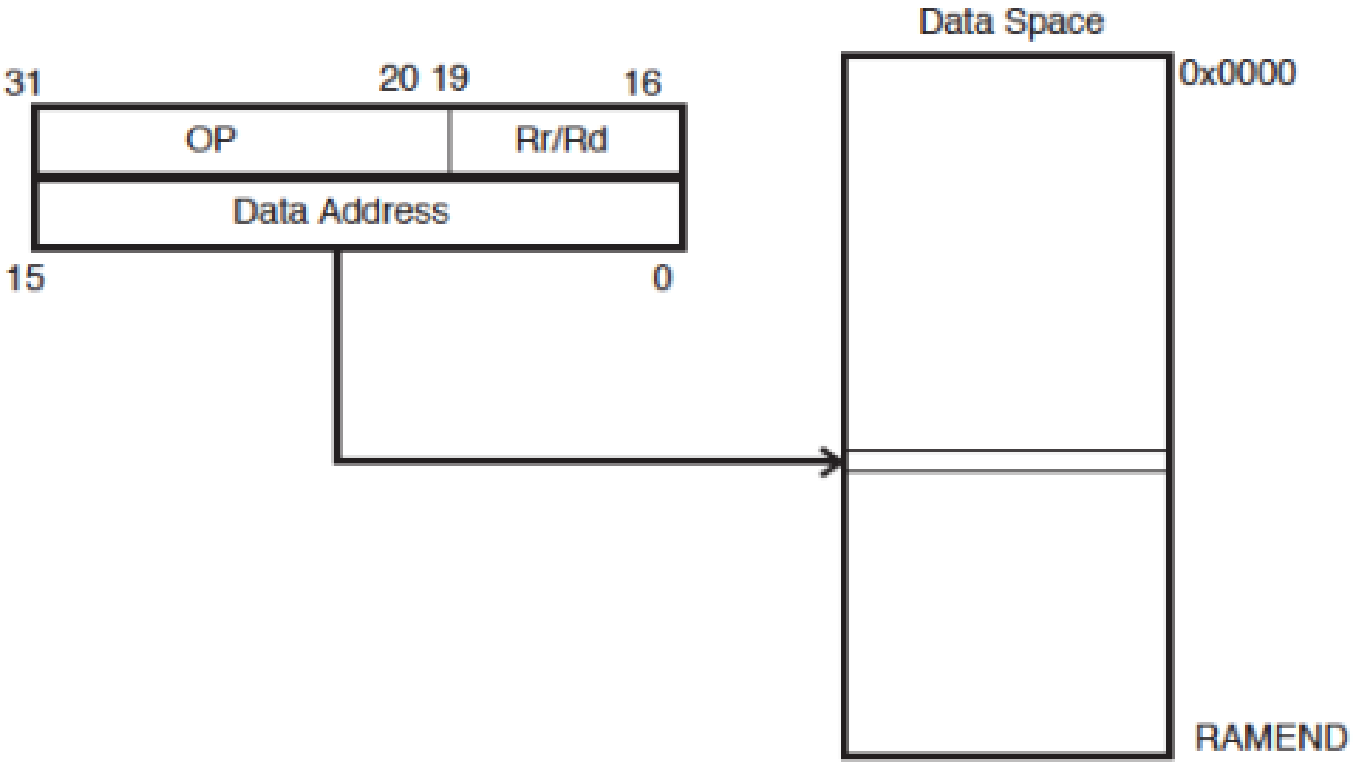
Modos de direccionamiento

- Directo a registro de E/S



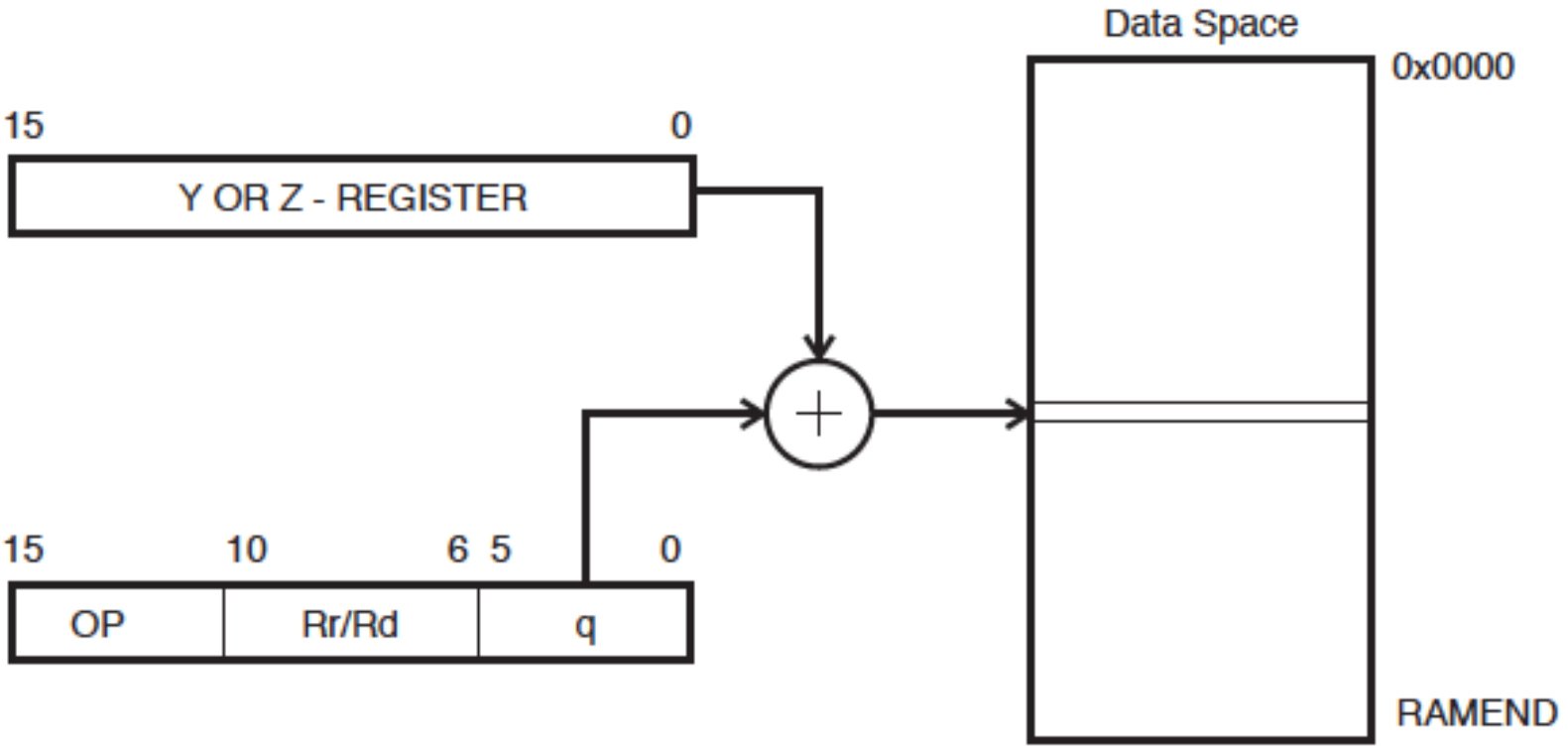
Modos de direccionamiento

- Directo a memoria de datos



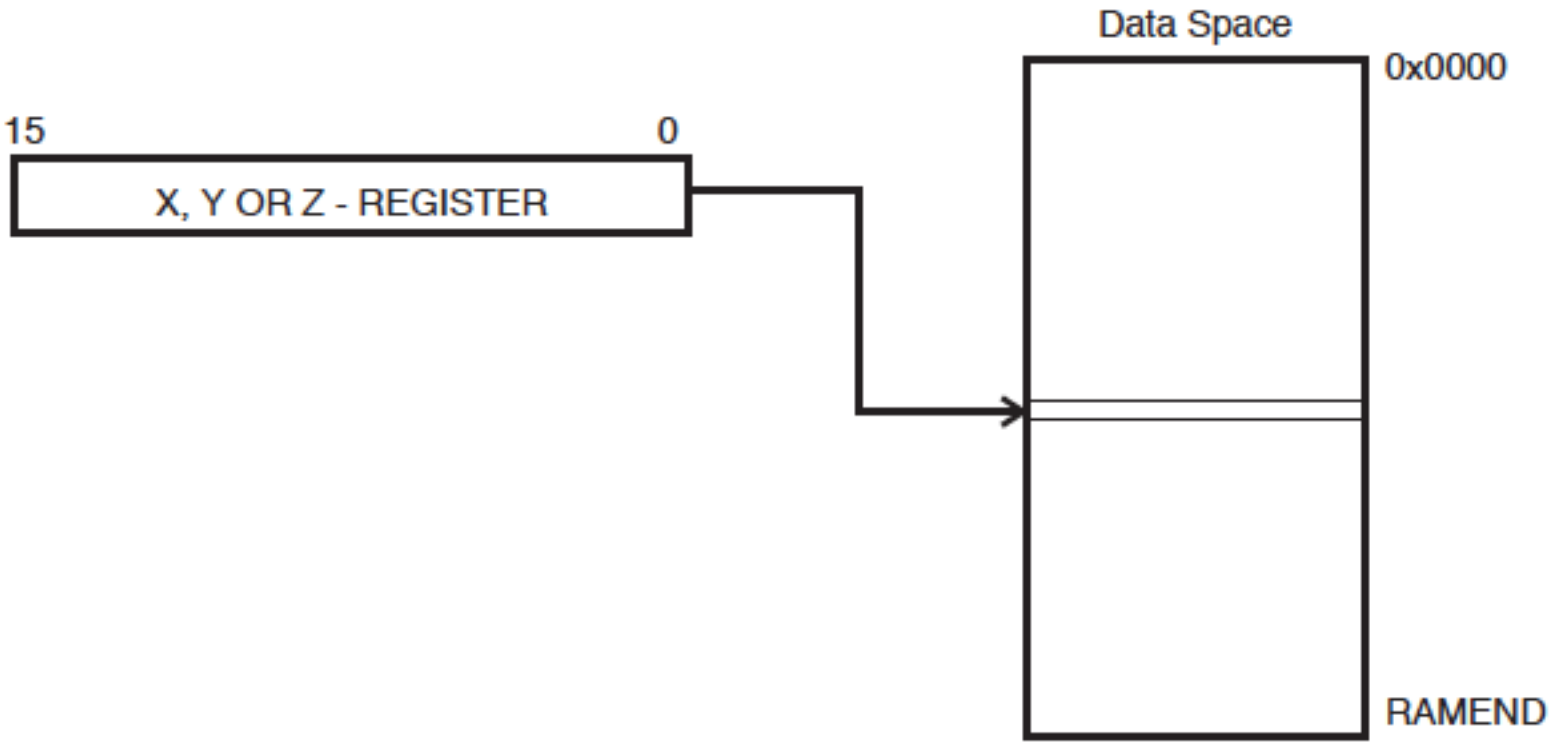
Modos de direccionamiento

- Directo a memoria de datos con desplazamiento



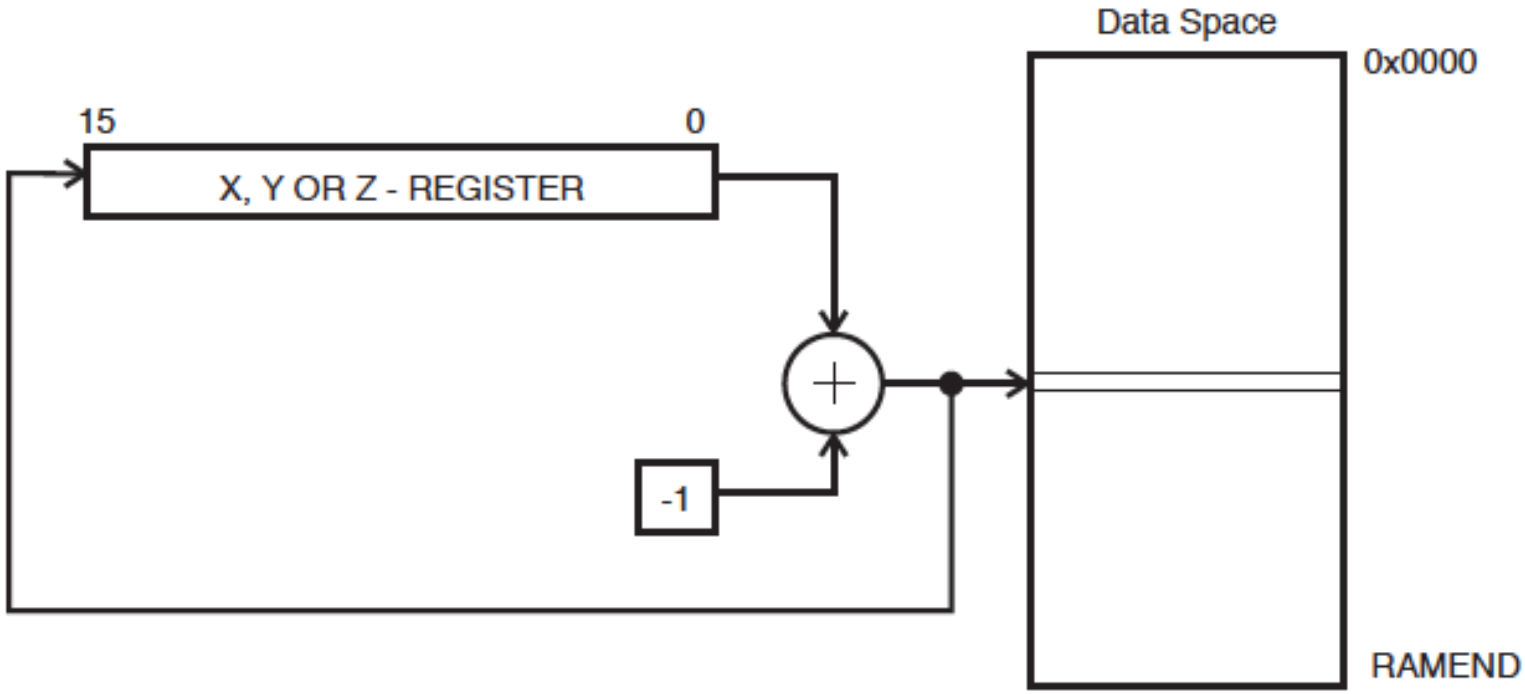
Modos de direccionamiento

- Indirecto a memoria de datos



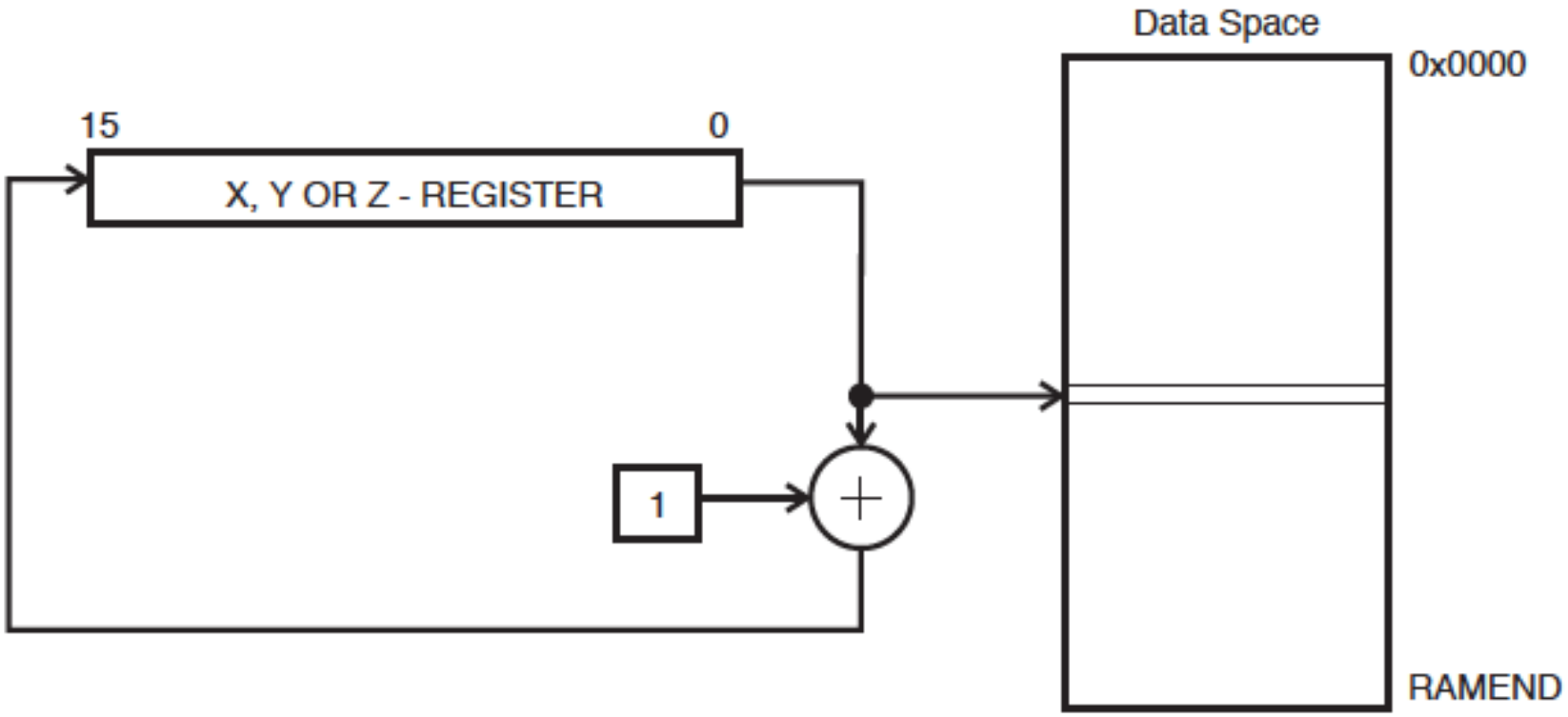
Modos de direccionamiento

- Indirecto a memoria de datos con predecremento



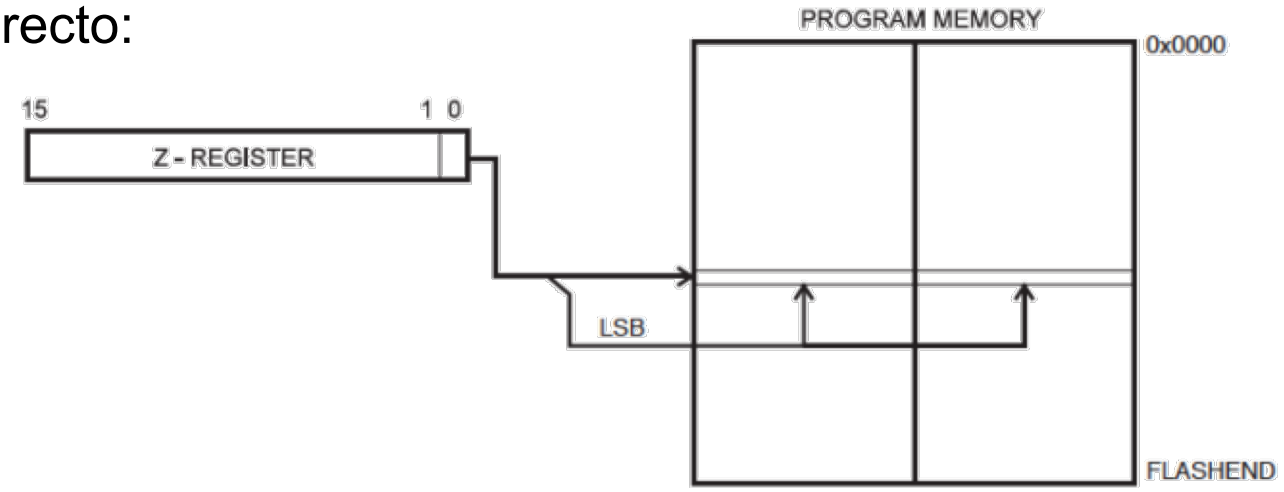
Modos de direccionamiento

- Indirecto a memoria de datos con postincremento



Modos de direccionamiento

- Acceso a constantes en memoria de programa
  - Indirecto:



  - Indirecto con postincremento: