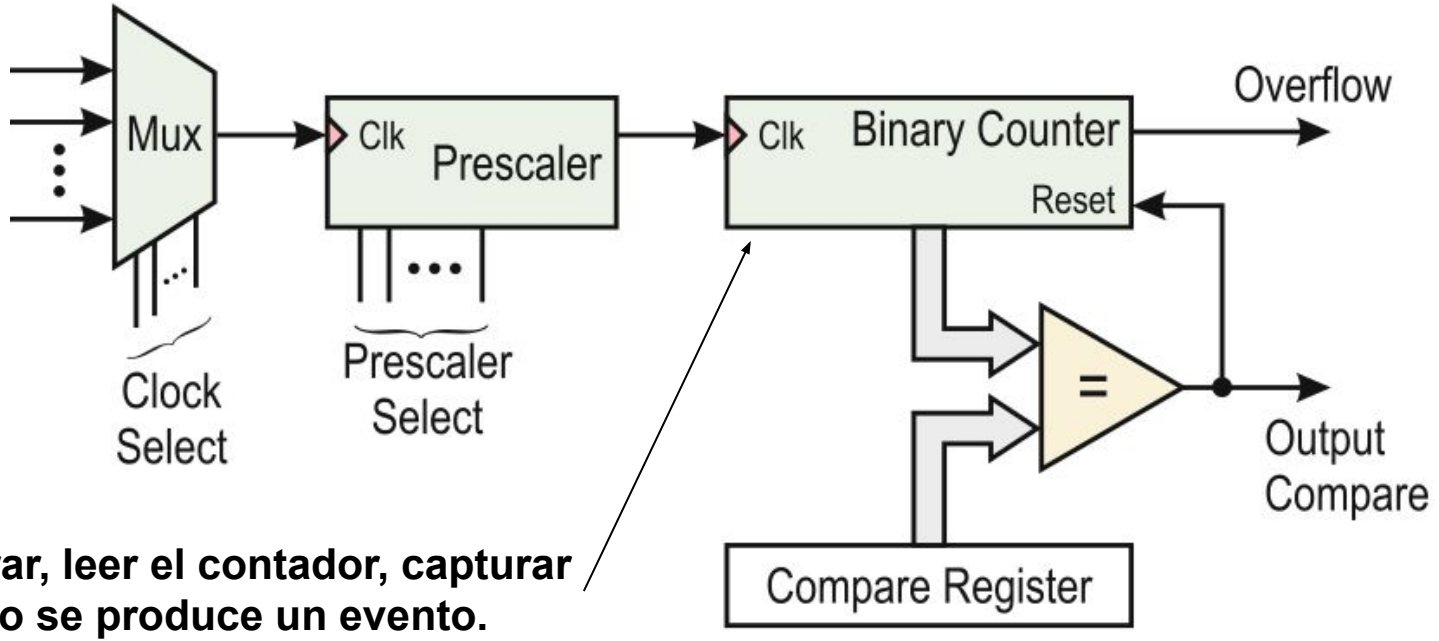
Sistemas Embebidos

Unidad 3

Timers - Interrupciones

Timer: esquema general con registro comparador

Reloj del procesador
Reloj externo
Pulsos en un pin



Se puede: comparar, leer el contador, capturar el contador cuando se produce un evento.

Figura obtenida de Jiménez et. al. “Introduction to Embedded Systems Using Microcontrollers and the MSP430”. Página 331

Temporizadores (Timers)

- Temporizadores (generar alguna señal a intervalos regulares de tiempo).
- Contadores de eventos.
- Relojes de tiempo real.
  - Pueden generar eventos temporizados con muy bajo error.
- Generación de velocidad en baudios.
- Modulación de ancho de pulso (PWM).
- Mediciones de tiempo entre dos eventos.
- Temporizadores watchdog.
- Generar interrupciones.
- Despertar al microcontrolador cuando está en modo bajo consumo

Timer: esquema general sin registro comparador

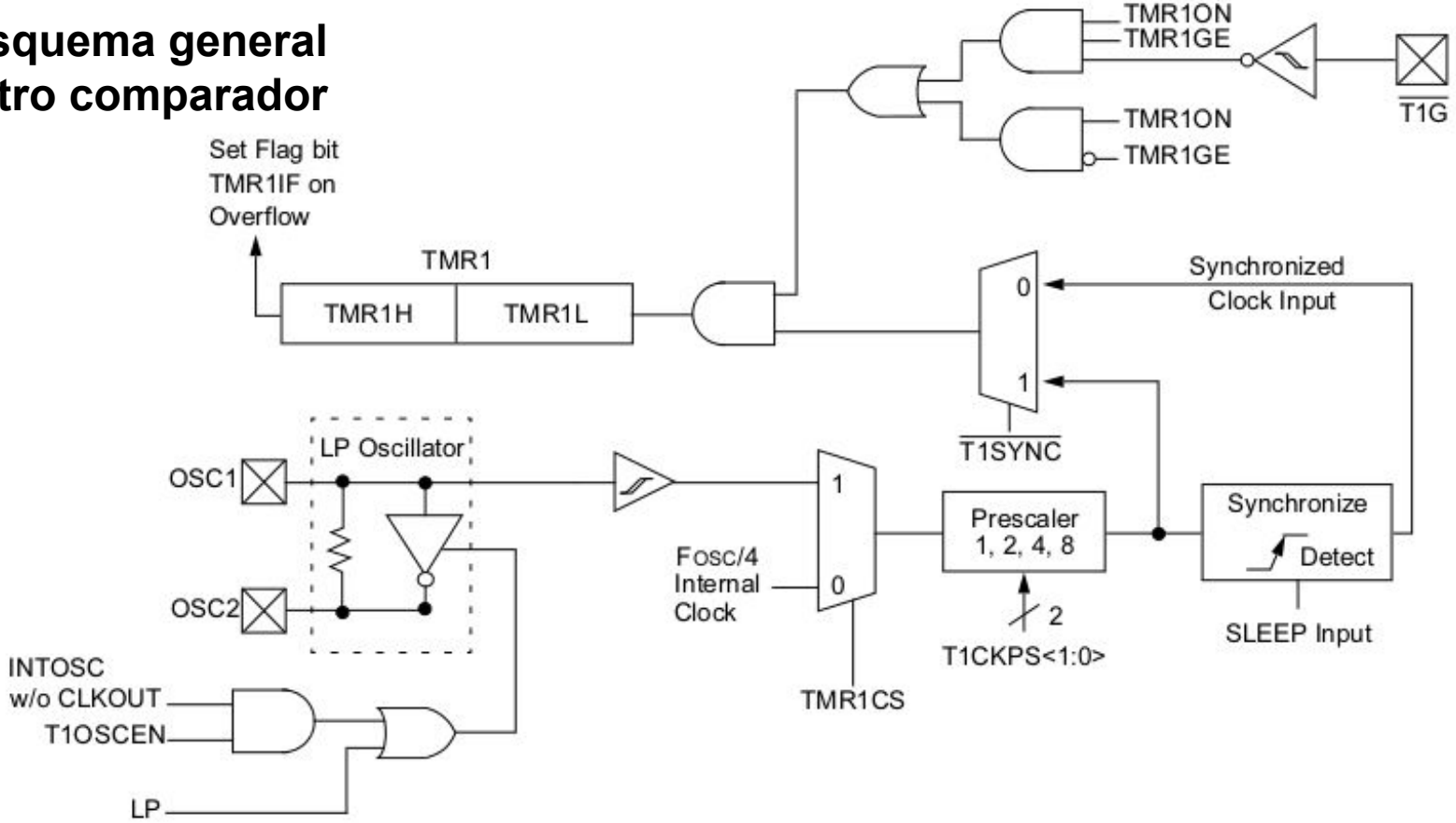
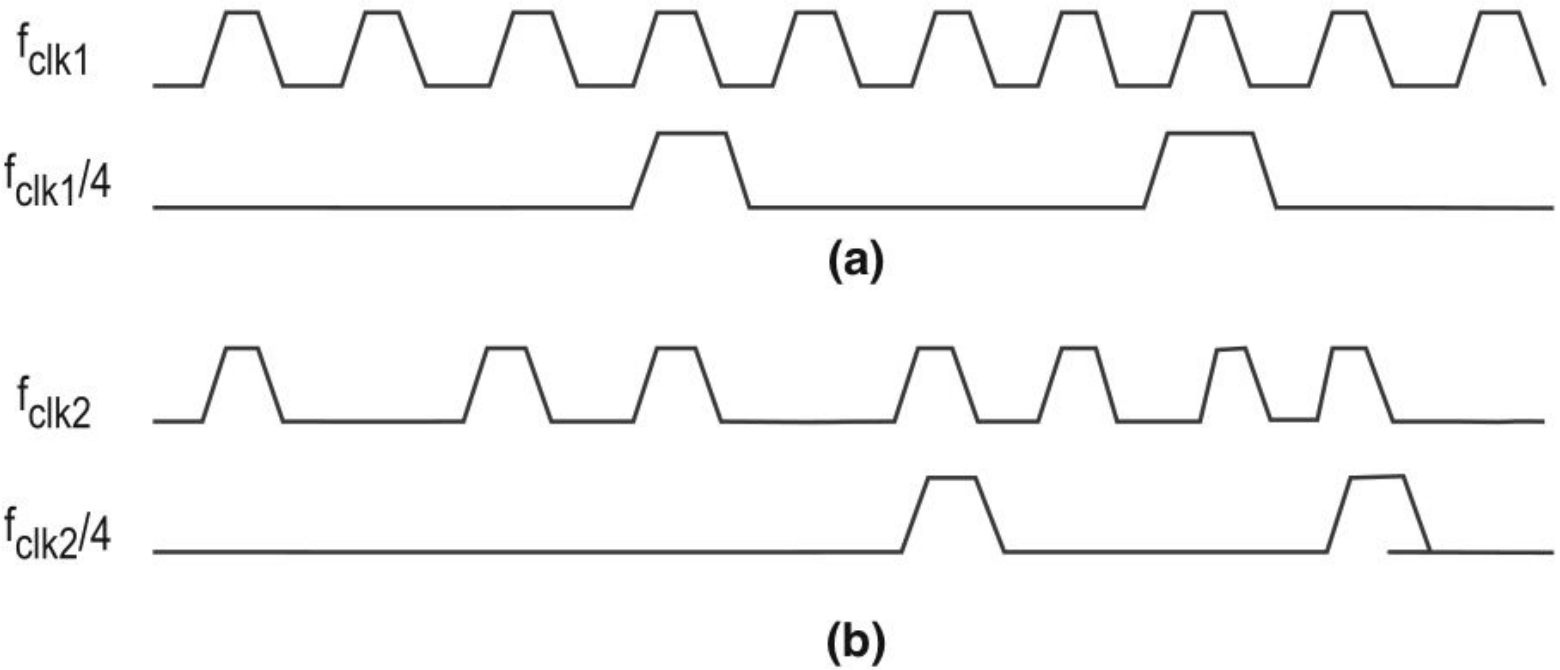


Figura obtenida de Microchip Technology Inc. PIC12F629/675 Data Sheet. Página 30

Timer: esquema general

- Preescaler: permite dividir la frecuencia de entrada.
- Binary counter: Registro que incrementa su valor por cada evento de reloj.
  - Puede arrancar de 0 o de otro valor.
- Compare register: Registro donde se escribe el valor máximo que alcanzará el timer.
- Fuentes de clock (clock sources):
  - Reloj del sistema.
  - Directamente del cristal.
  - Señal en un pin (no necesariamente periódica).

Funcionamiento como temporizador y como contador de eventos

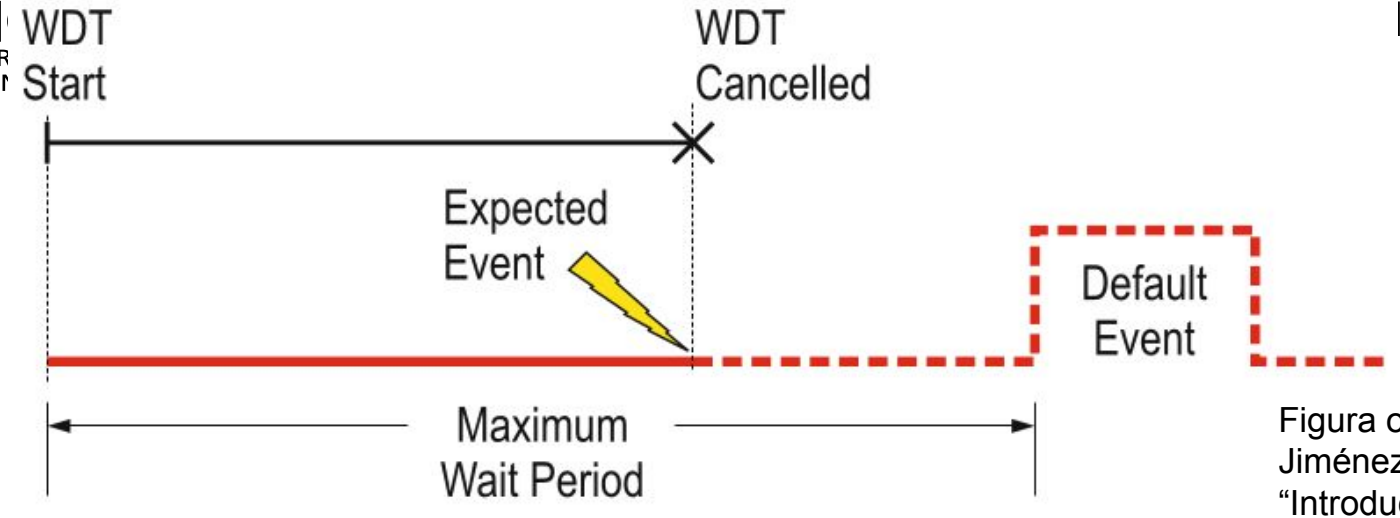


a) Temporizador; b) Contador de eventos

Watchdog Timers

- Sistema de seguridad que produce un **evento por defecto** si un **evento esperado no ocurre** durante un intervalo de tiempo definido.
  - Salir de lazos infinitos.
  - Salir de programas con comportamientos no pronosticados.
- Ejemplo común:
  - Resetear el microcontrolador si el programa no pasa por puntos predefinidos.
  - Cambiar de estado una salida si un evento externo no ocurre (apagar un sistema en una condición crítica si el usuario no lo apaga en un intervalo de tiempo específico).

e la



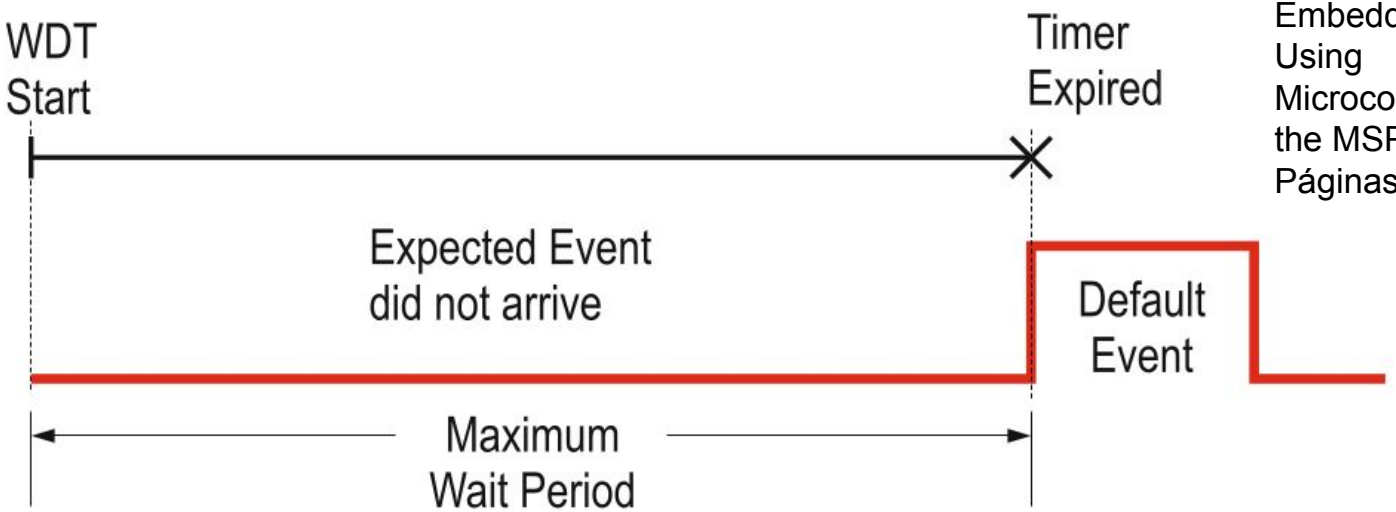
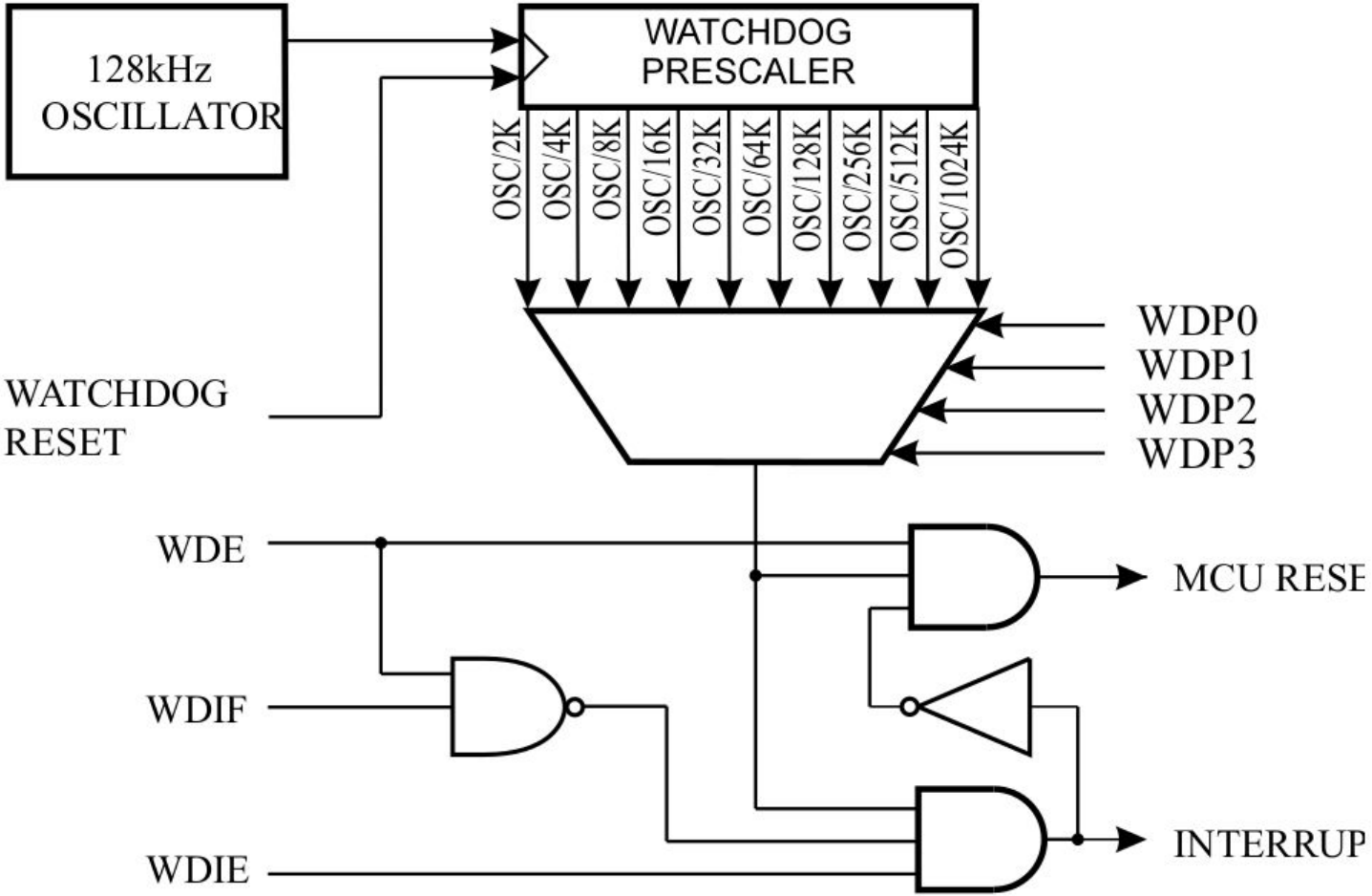


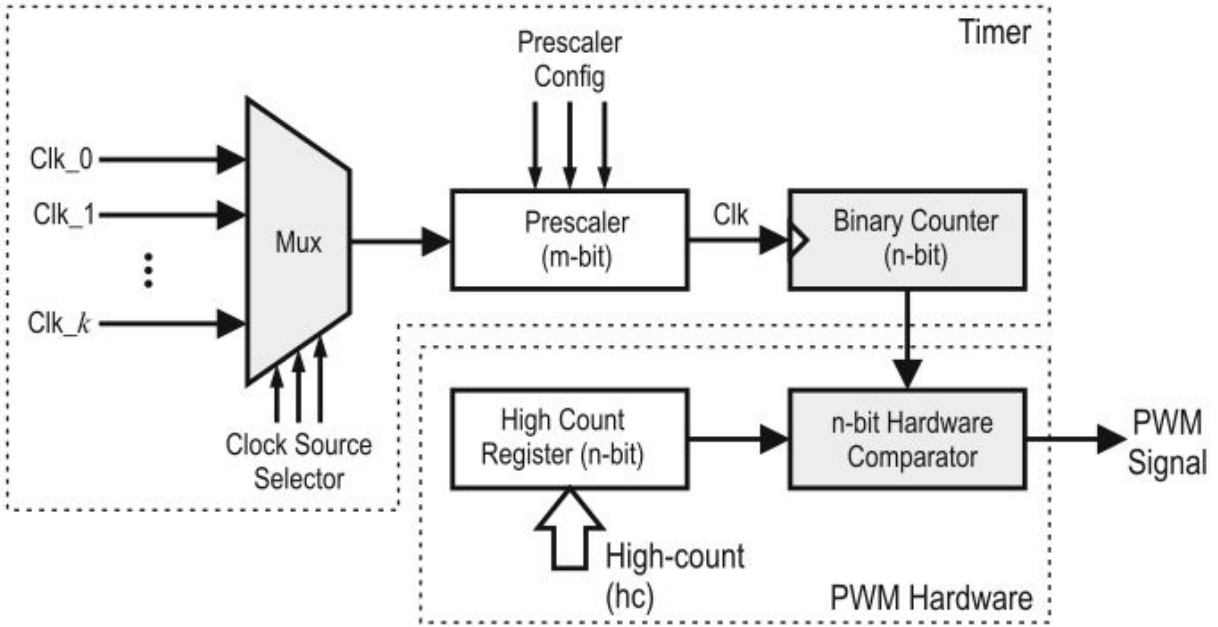
Figura obtenida de Jiménez et. al. “Introduction to Embedded Systems Using Microcontrollers and the MSP430”. Páginas 336 y 337

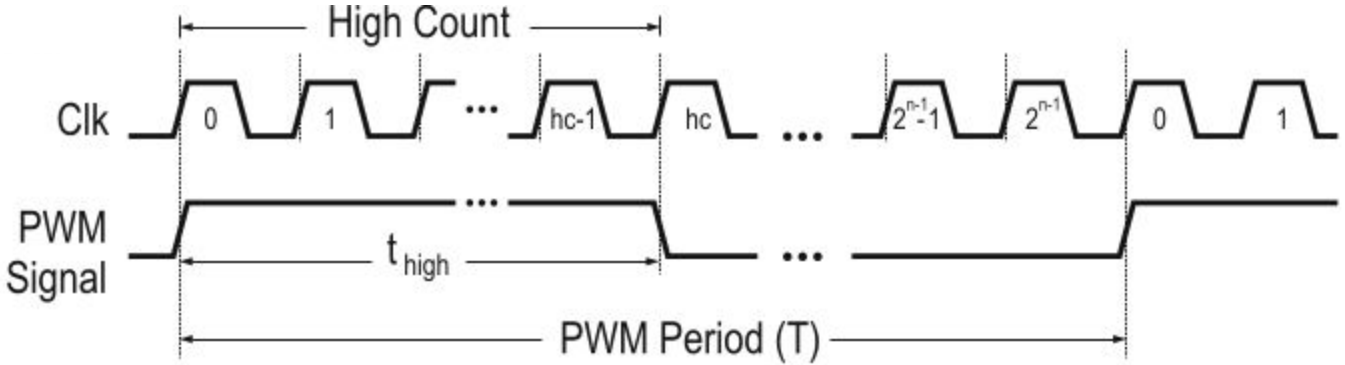
Watchdog Timers



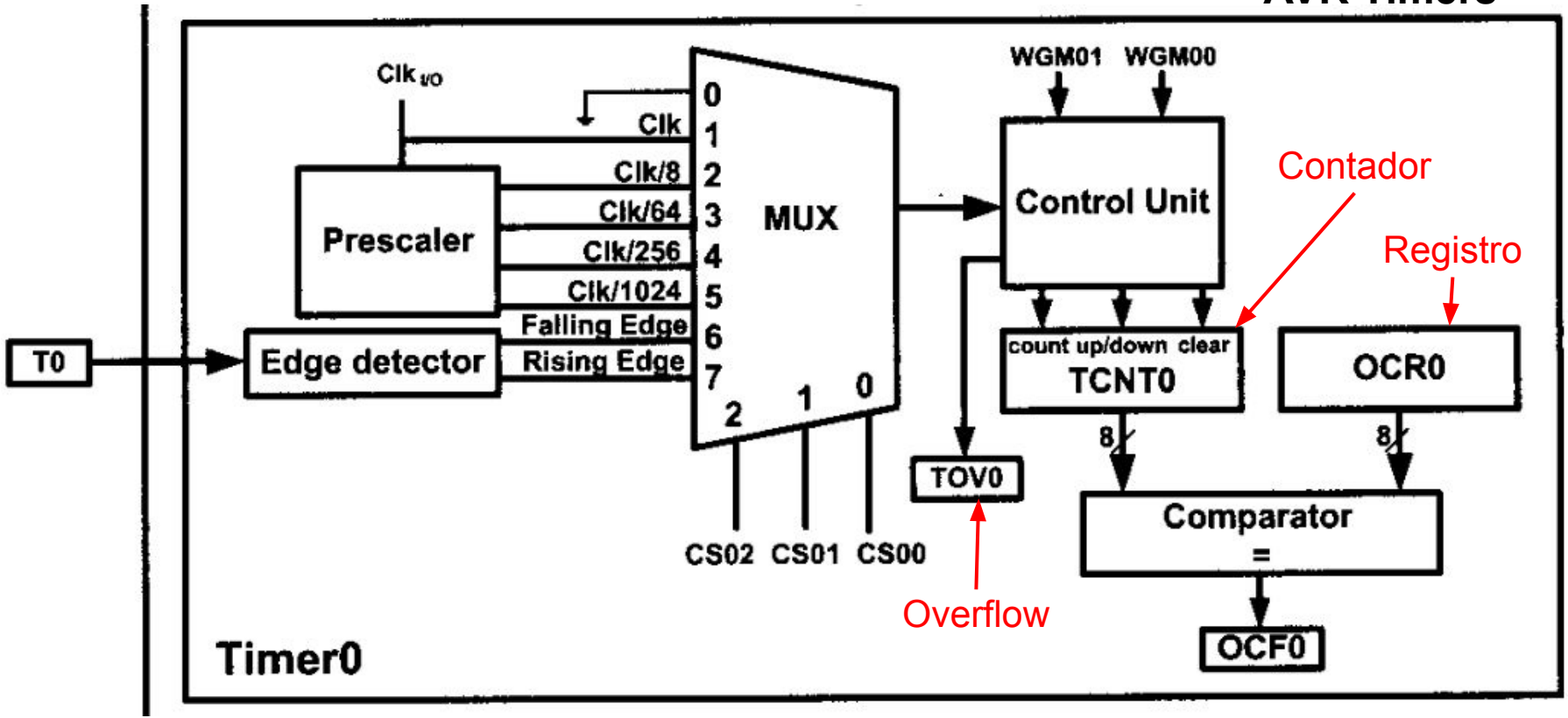
Figuras obtenida de Atmel, “8-bit AVR Microcontrollers. ATmega328-328P Datasheet complete”. Páginas 76

PWM


Timer
Prescaler Config
Clk_0
Clk_1
Clk_k
Mux
Prescaler (m-bit)
Clk
Binary Counter (n-bit)
Clock Source Selector
High Count Register (n-bit)
n-bit Hardware Comparator
PWM Signal
High-count (hc)
PWM Hardware

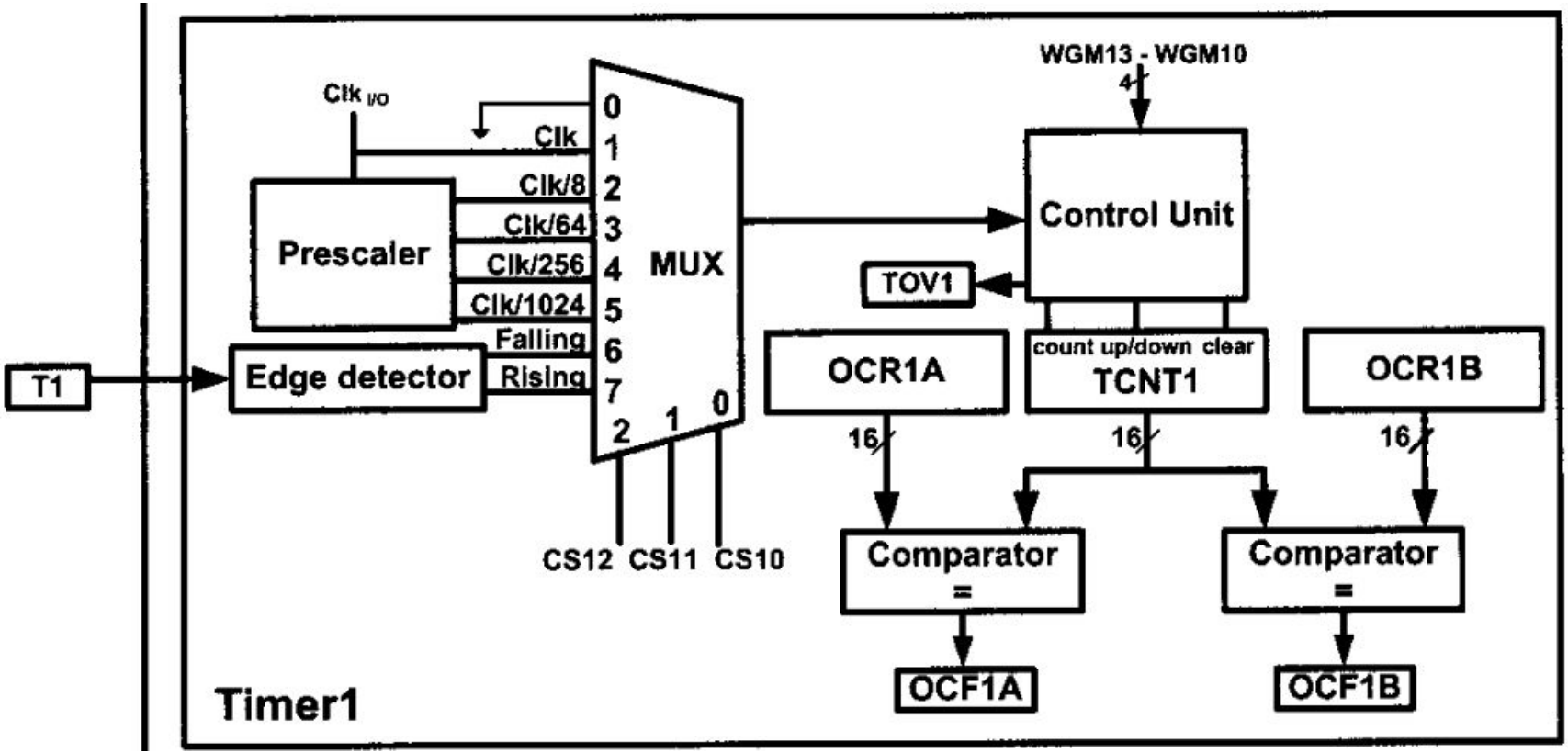

High Count
Clk
0
1
hc-1
hc
2^n-1
2^n-1
0
1
PWM Signal
t high
PWM Period (T)

AVR Timers



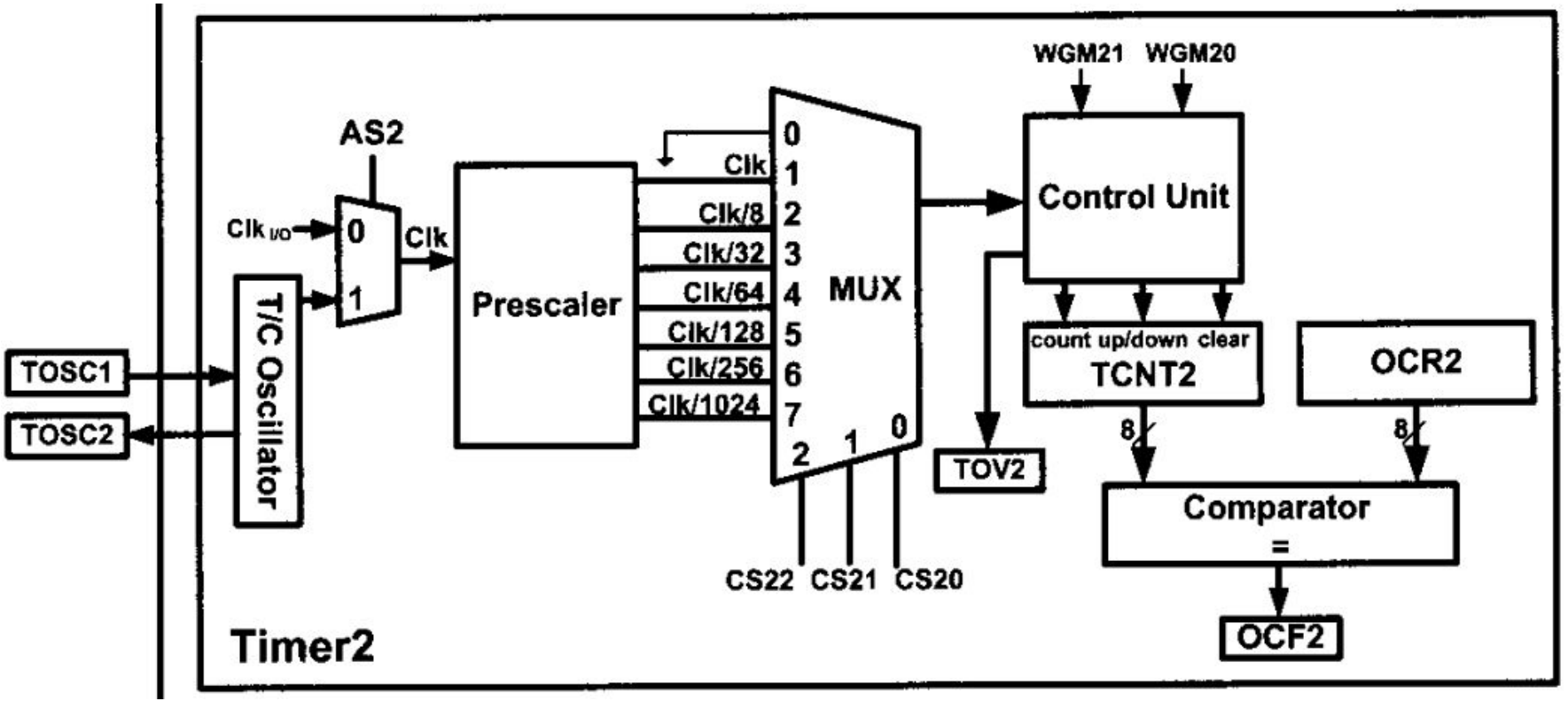
Mazidi et. al. "The AVR microcontroller and embedded system: using Assembly and C". Página 312.

AVR Timers



Mazidi et. al. "The AVR microcontroller and embedded system: using Assembly and C". Página 312.

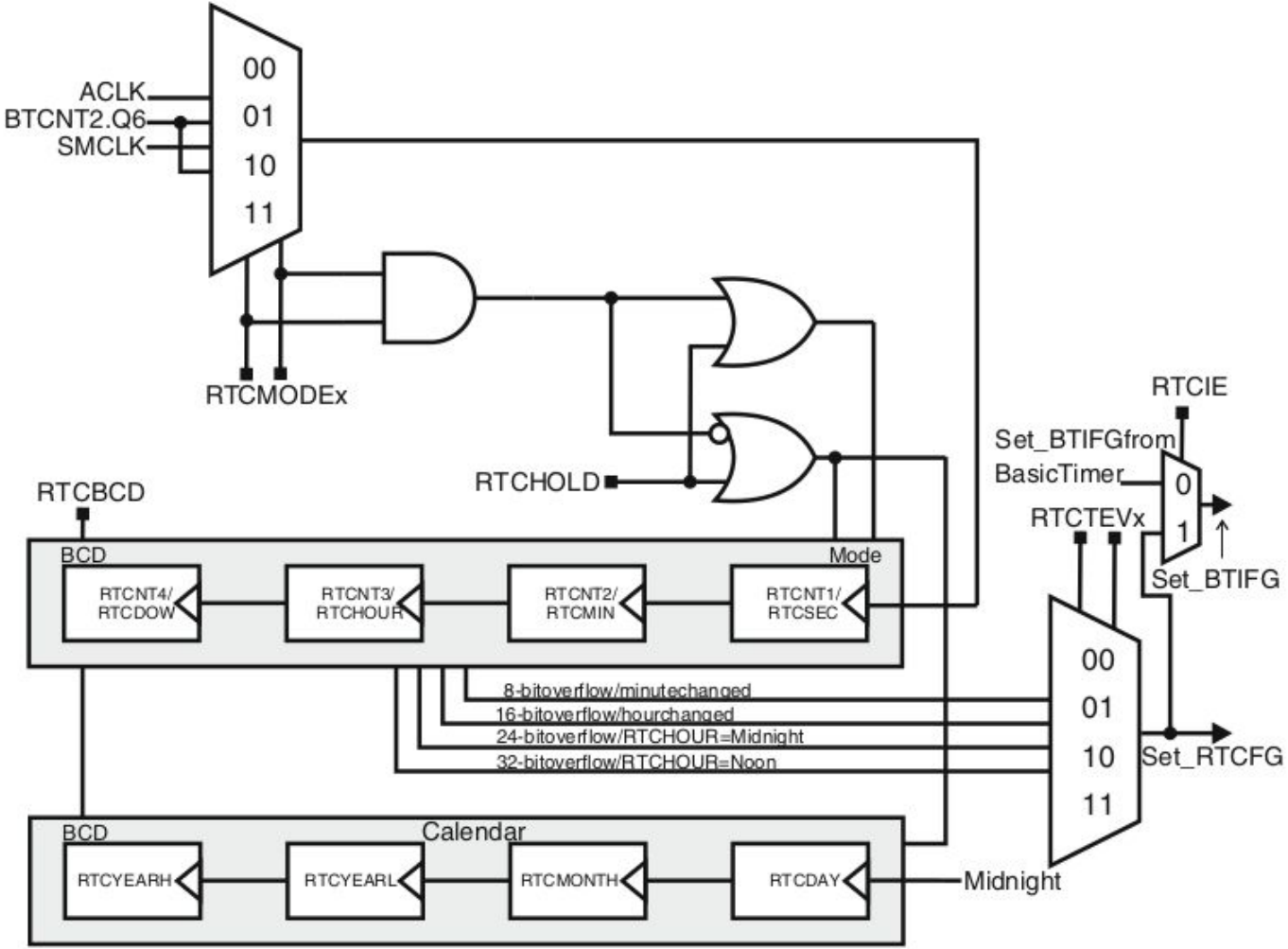
AVR Timers



Mazidi et. al. "The AVR microcontroller and embedded system: using Assembly and C". Página 312.

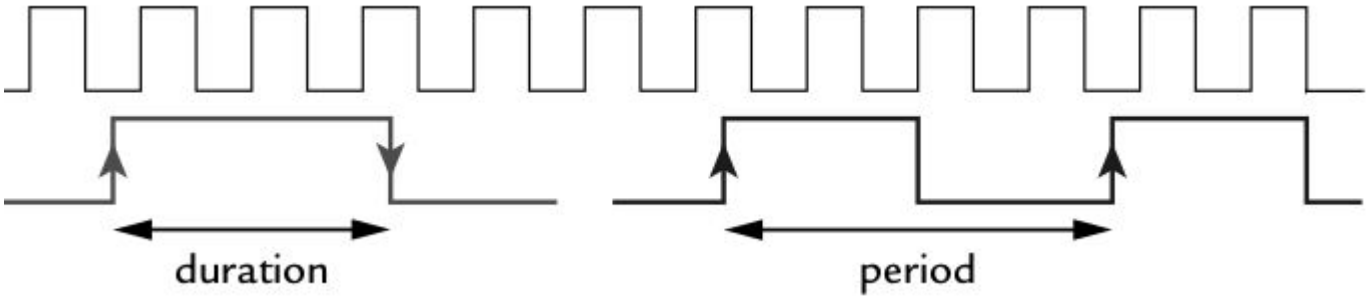
Real time Clock

- Temporizador que cuenta segundos, minutos, horas, días, meses, años, siglos, etc.
  - Los datos son mantenidos en registros.
  - Pueden generar interrupciones periódicas (cada segundo, cada minuto, etc.)


00
01
10
11
ACLK
BTCNT2.Q6
SMCLK
RTCMODEx
RTCHOLD
RTCBCD
BCD
Mode
RTCNT4/
RTCDOW
RTCNT3/
RTCHOUR
RTCNT2/
RTCMIN
RTCNT1/
RTCSEC
8-bitoverflow/minutechanged
16-bitoverflow/hourchanged
24-bitoverflow/RTCHOUR=Midnight
32-bitoverflow/RTCHOUR=Noon
RTCIE
Set_BTIFGfrom
BasicTimer
RTCTEVx
0
1
Set_BTIFG
Set_RTCFG
BCD
Calendar
RTCYEARH
RTCYEARL
RTCMONTH
RTCDAY
Midnight

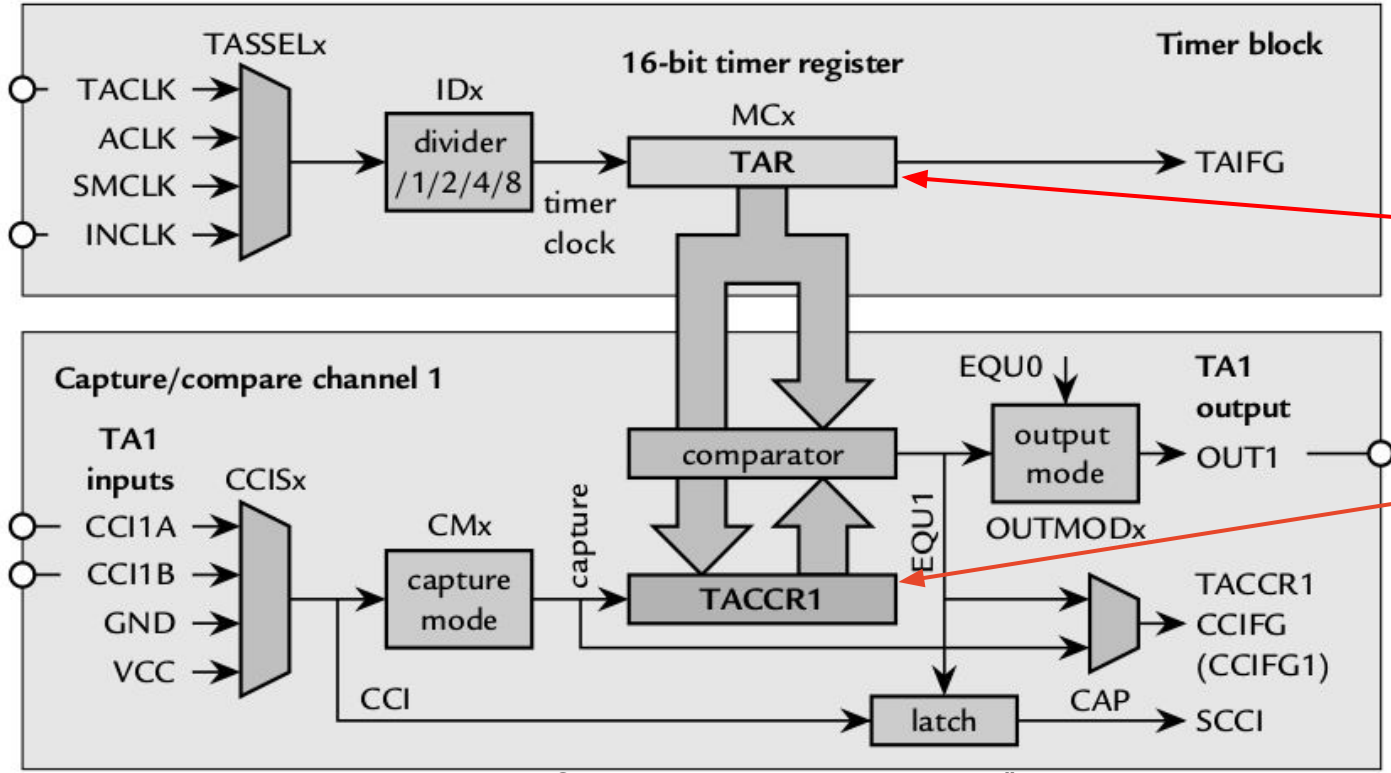
Medición de intervalos de tiempo - Capture compare

- Objetivo:
  - Medir tiempo entre dos eventos. Usualmente un evento en un pin.
    - Tiempo entre dos flancos del mismo tipo (ejemplo, dos flancos de subida)..
    - Tiempo entre dos flancos cualquiera.



Medición de intervalos de tiempo - Capture compare

Permite agregar estampas de tiempo a eventos + funcionar como comparador.



Timer

Registro captura. Ante una señal de entrada, el valor del timer pasa al registro de captura

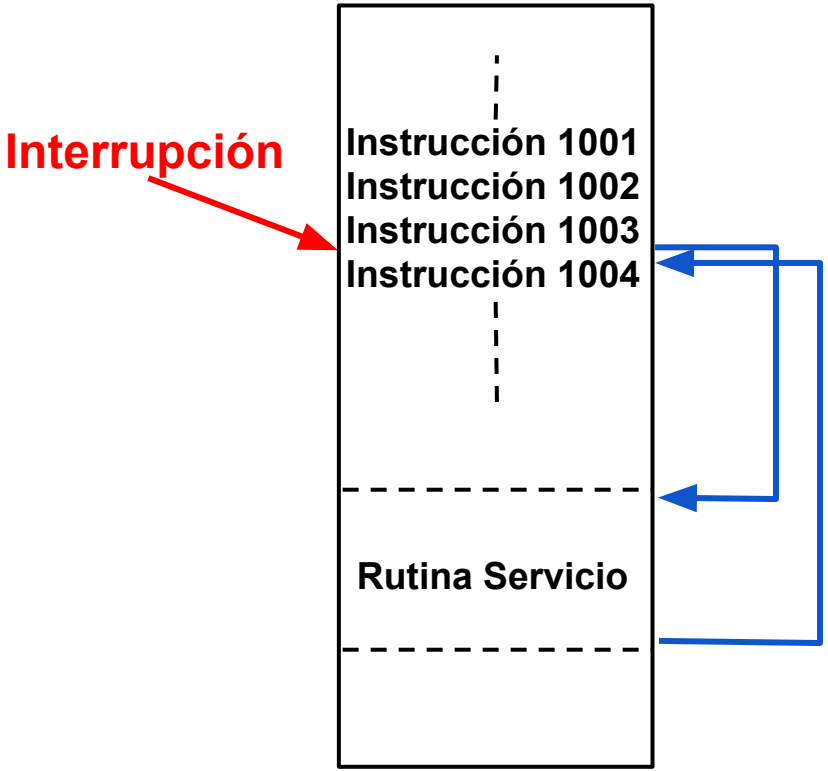
Figura obtenida de John Davies, "MSP430 Microcontroller Basics". Elsevier. 2008. Página 312.

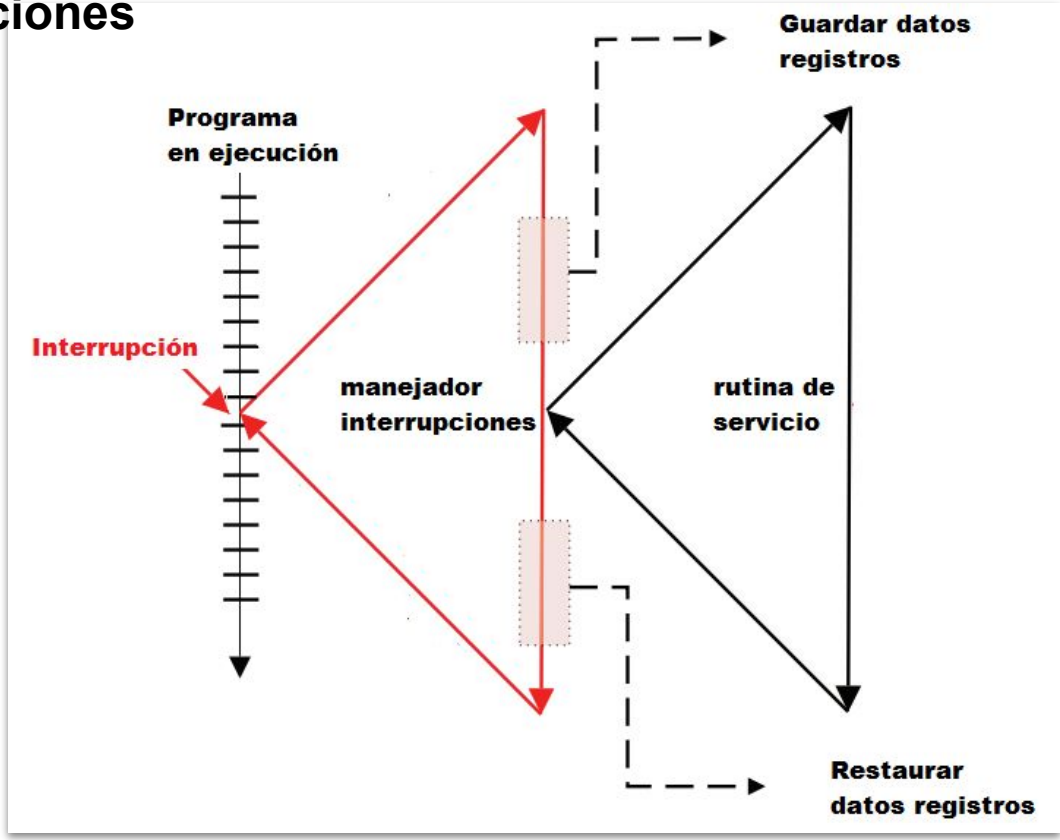
Ejemplo: BCM2835

- 4 timers de 32 bits + 1 contador de 64 bits.
  - Los 4 timers comparan el contenido de un registro de comparación con los 32 bits menos significativos del contador.
  - Cuando los valores coinciden, se produce una interrupción.

ST Address Map			
Address Offset	**Register Name**	**Description**	**Size**
0x0	CS	System Timer Control/Status	32
0x4	CLO	System Timer Counter Lower 32 bits	32
0x8	CHI	System Timer Counter Higher 32 bits	32
0xc	C0	System Timer Compare 0	32
0x10	C1	System Timer Compare 1	32
0x14	C2	System Timer Compare 2	32
0x18	C3	System Timer Compare 3	32

Interrupciones





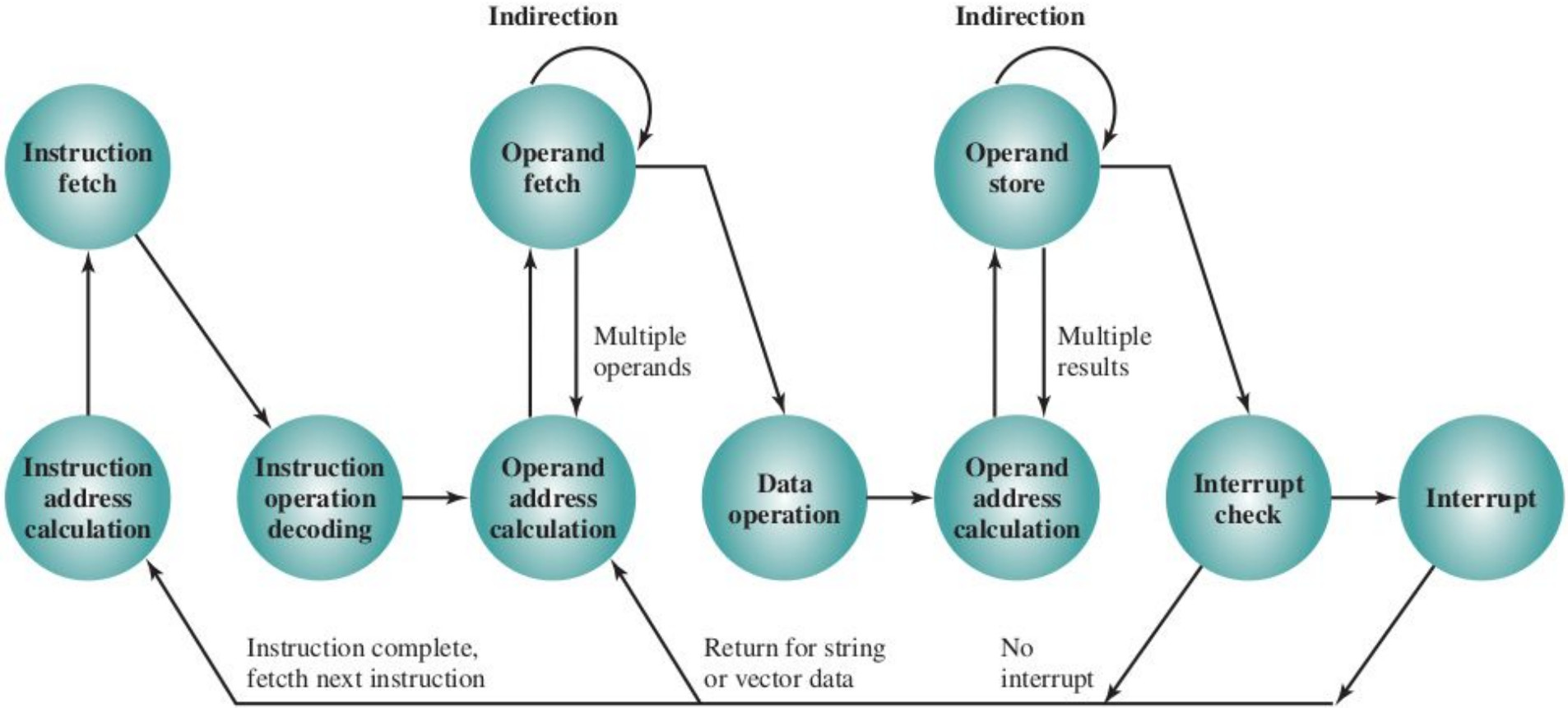


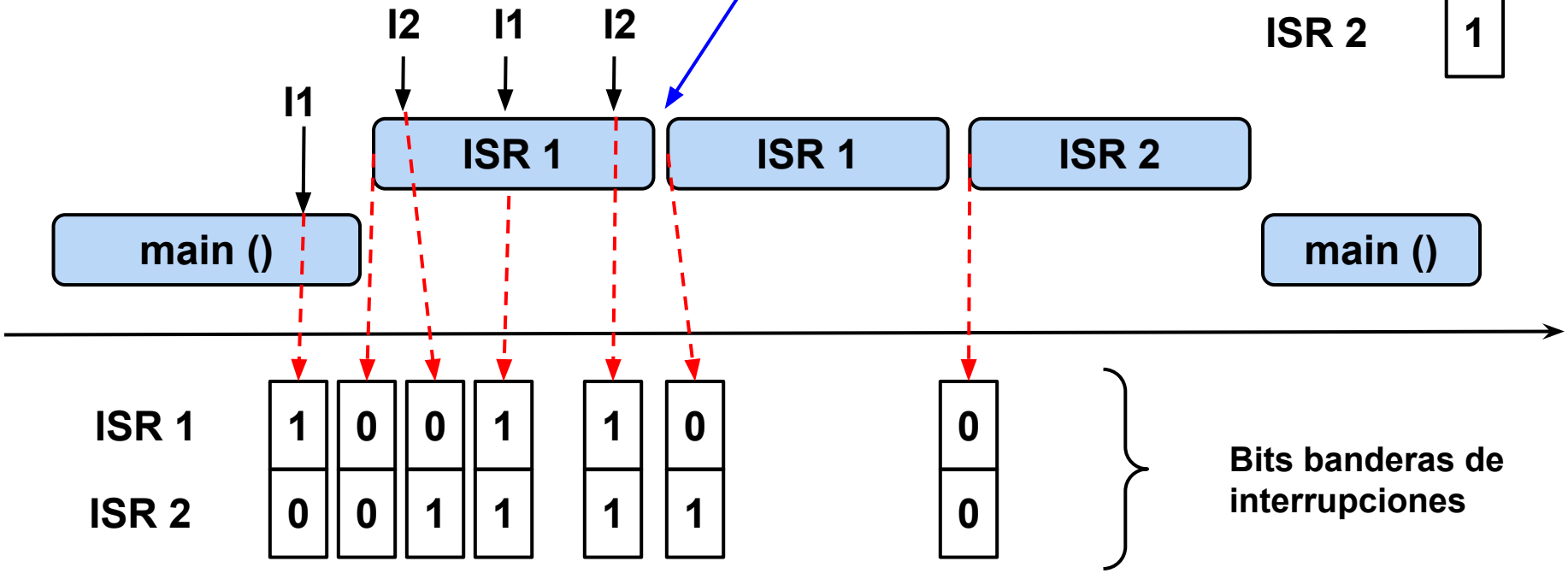
Figura obtenida de Computer Organization and Architecture 9th Edition William Stallings, página 493

Non-maskable interrupts:
Interrupciones que no pueden

La interrupción 1 tiene prioridad

Bits máscara de interrupciones

ISR 1	1
ISR 2	1



ISR 1	1	0	0	1	1	0	0		
ISR 2	0	0	1	1	1	1	0		

Bits banderas de interrupciones

Vector No	Program Address(2)	Source	Interrupts definition
1	0x0000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 0
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/Coutner2 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

Figura obtenida de Atmel, “8-bit AVR Microcontrollers. ATmega328-328P Datasheet complete”. Página 82

13	0x0018	TIMER1_COMPB	Timer/Coutner1 Compare Match B
14	0x001A	TIMER1_OVF	Timer/Counter1 Overflow
15	0x001C	TIMER0_COMPA	Timer/Counter0 Compare Match A
16	0x001E	TIMER0_COMPB	Timer/Coutner0 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

Figura obtenida de Atmel, “8-bit AVR Microcontrollers. ATmega328-328P Datasheet complete”. Página 82

External Interrupt Mask Register

Name: EIMSK
Offset: 0x3D
Reset: 0x00
Property: When addressing as I/O Register: address offset is 0x1D

Bit	7	6	5	4	3	2	1	0
							INT1	INT0
Access							R/W	R/W
Reset							0	0

External Interrupt Flag Register

Name: EIFR
Offset: 0x3C
Reset: 0x00
Property: When addressing as I/O Register: address offset is 0x1C

Bit	7	6	5	4	3	2	1	0
							INTF1	INTF0
Access							R/W	R/W
Reset							0	0

Figura obtenida de Atmel, “8-bit AVR Microcontrollers. ATmega328-328P Datasheet complete”. Página 91

Interrupciones en alto nivel

- En lenguajes de alto nivel las rutinas de servicio se escriben como funciones (o subrutinas) callback
  - Serán llamadas cuando se produzca la interrupción.
  - Deben ser escritas por el programador

```
void ISR_pin2(){
  println("Rutina de servicio");
  for(int i=1;i<8;i++){
  ..............................
  ..............................
}
```

call ISR_pin2() 

[1]Función que se pasa como un argumento a otra función. Se invoca dentro de la función externa para completar alguna tarea.

Modos bajo consumo

- Permiten bajar el consumo de energía mediante diferentes técnicas.
- Los modos de bajo consumo combinan una o varias de estas técnicas.
- Las técnicas más usuales son:
  - Deteniendo los relojes que controlan a diferentes periféricos o al procesador.
    - Se reduce el consumo de energía moderadamente.
  - Apagando periféricos que no están en uso o el procesador (quitando la energía eléctrica).
    - Se reduce el consumo de energía profundamente.
  - Bajando la frecuencia de operación.
  - Bajando el voltaje de operación.
- En lenguaje ensamblador: Instrucción SLEEP + varios bits de configuración.

Modos de Bajo consumo comunes

Modo	Apaga/detiene	Sigue funcionando	Lo despierta	Consumo
Idle Mode	Detiene relojes del procesador y memoria flash.	El resto de los periféricos.	Interrupción interna o externa.	1/5 a 1/10 consumo normal
ADC Noise Reduction Mode	Detiene relojes del procesador, memoria flash y GPIO. ADC encendido.	Timers, ADC, interrupciones en pines.	Interrupciones de Timers, ADC y pines.	Menor que modo Idle
Power-Down Mode	Quita energía a casi todo el micro	interrupciones externas, Watchdog, puerto serie	Interrupción externa, Watchdog, Brown-out Reset, puerto serie	2-15 µA

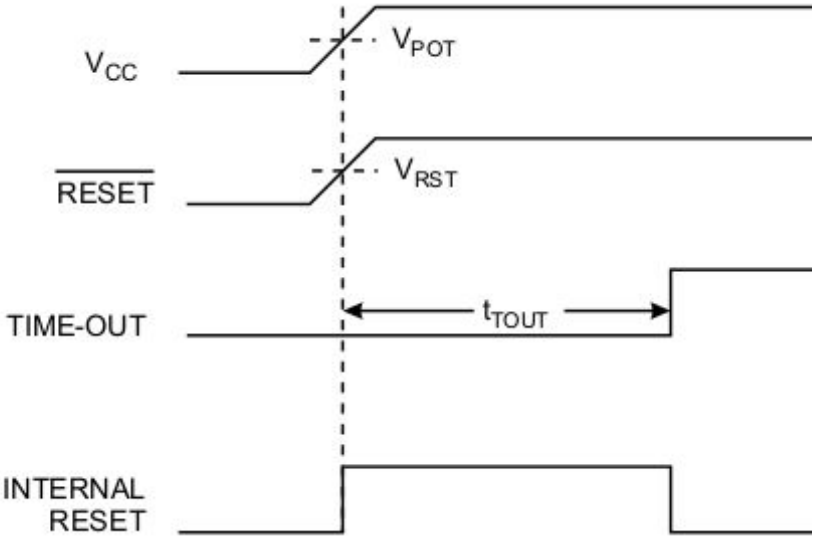
Modos de Bajo consumo comunes

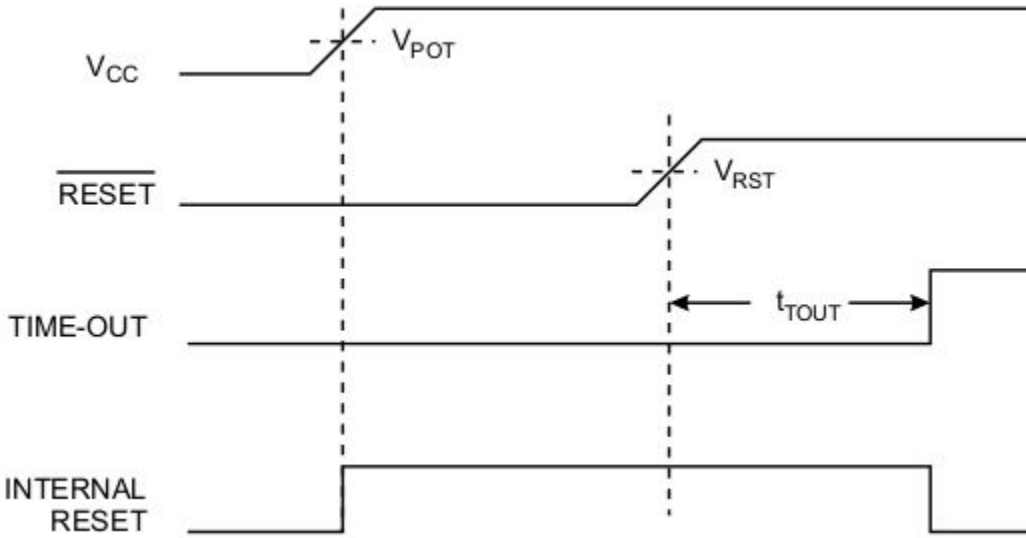
Modo	Apaga/detiene	Sigue funcionando	Lo despierta	Consumo
Power-save Mode	Todo el micro, menos timer 2.	interrupciones externas, Watchdog, puerto serie, timer 2	Interrupción externa, Watchdog, Brown-out Reset, puerto serie, timer 2	menos que 2-15 µA
Standby Mode	Todo el micro	oscilador externo	el oscilador después de 6 ciclos	menos que 1 µA
Extended Standby Mode	Todo el micro	oscilador externo	el oscilador después de 6 ciclos	menos que 1 µA

Modos de Reset

- Diferentes eventos pueden producir un reset.
  - Tensión de alimentación cae por debajo de un umbral.
  - Tensión de alimentación fluctuando dentro de un umbral que puede causar mal funcionamiento (Brown-out Reset).
  - Se genera una señal en un pin RESET.
  - Watchdog timer.
- El procesador no vuelve a funcionar hasta que se cumplen condiciones especificadas.

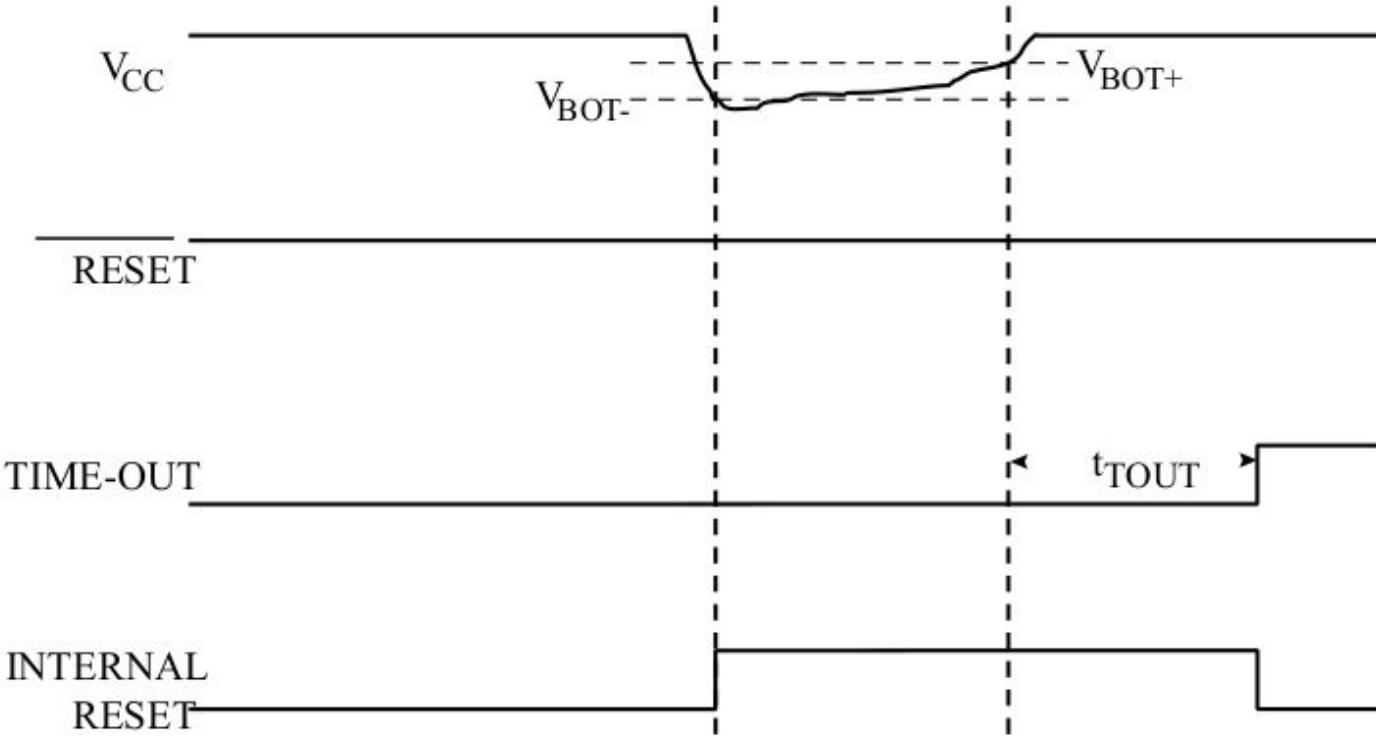
Modos de Reset
V_{CC} y señal RESET





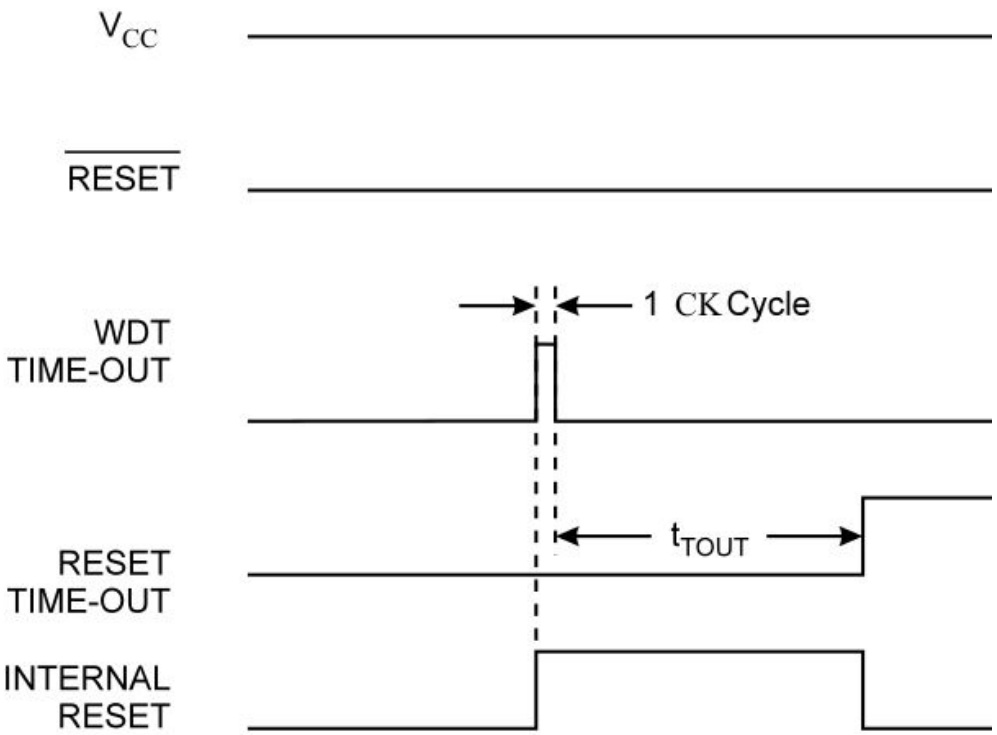
Figuras obtenida de Atmel, “8-bit AVR Microcontrollers. ATmega328-328P Datasheet complete”. Páginas 73 y 74

Modos de Reset: Brown-out Reset



Figuras obtenida de Atmel, “8-bit AVR Microcontrollers. ATmega328-328P Datasheet complete”. Páginas 75

Modos de Reset: Watchdog timer

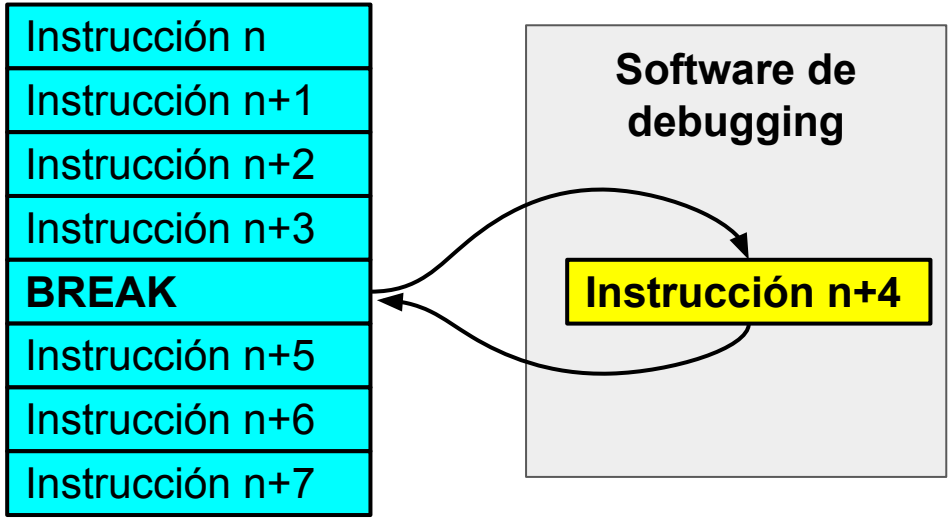


Figuras obtenida de Atmel, “8-bit AVR Microcontrollers. ATmega328-328P Datasheet complete”. Páginas 75

Mecanismos de debugging

- Mecanismo que permite realizar debugging utilizando algunas facilidades del microcontrolador.
  - Interfaz al hardware real.
  - Usualmente comunicación serial a través de conectores estándares como:
    - JTAG (Joint Test Action Group).
    - BDM (Background debug mode)
- Permite inspeccionar:
  - El estado interno (registros de estado, registros de propósito general, posiciones de memoria, etc.).
  - Valores de variables.
  - Fijar checkpoints, breakpoints y watchpoints.

- **Breakpoints**: detienen el flujo de un programa.
  - Instrucción “break”.
  - Se inserta en lugar de la instrucción donde queremos detener el programa. La instrucción se reemplaza con “break”.
  - Durante debugging, el software de debugging almacena la instrucción.



- Checkpoint: Punto para verificar si el programa pasa por un punto, pero sin detener el programa.
- Watchpoints: apunta a datos. El flujo del programa se detiene si el dato apuntado cambia.

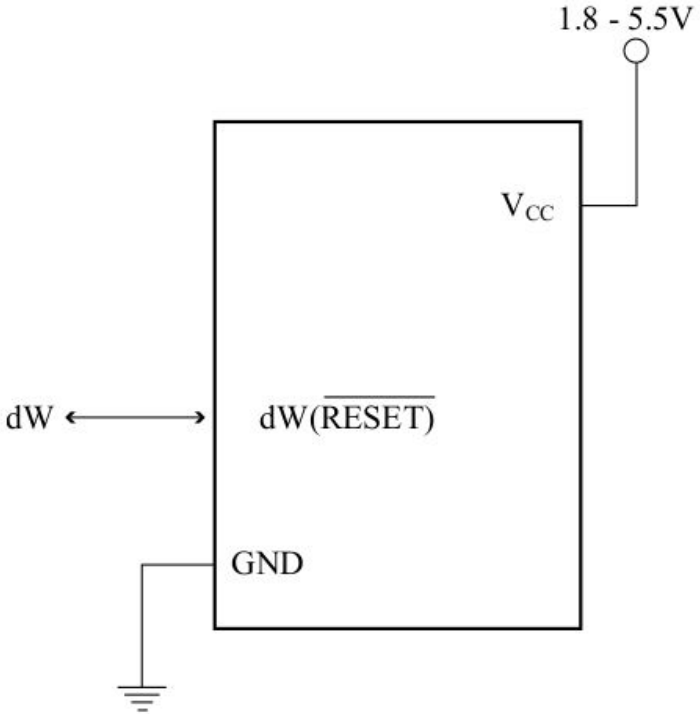
Mecanismos de debugging

- On-chip debugging:
  - Se implementa mediante circuitos electrónicos adicionales dentro del chip.
- In-circuit debugging:
  - Hardware adicional que se conecta entre el sistema embebido y la computadora + programa monitor que se ejecuta en el microcontrolador.
  - Microchip Technology

Mecanismos de debugging

- In-circuit emulation:
  - Permite emular parte del hardware del sistema embebido, especialmente el procesador, mediante software que interactúa con el resto del hardware real.
  - Basado en la generación de señales.
  - Ejemplo: JTAG estándar para In-circuit emulation.
- debugWIRE:
  - Microcontroladores de recursos limitados.
  - Usa un solo pin bidireccional
  - Microcontroladores AVR y PICs pequeños.
  - breakpoints, single step, emulación, etc.

Mecanismos de debugging
Ejemplo: debugWire de Atmel 328P



- Controlar el flujo del programa.
- Emular todas las funciones del microcontrolador, analógicas y digitales, menos el reset.
- Operaciones en tiempo real.
- Emulación en Ensamblador y en C.
- Break Points
- Emular características eléctricas idénticas a las del dispositivo real.
- Escribir memoria no volátil

Figuras obtenida de Atmel, “8-bit AVR Microcontrollers. ATmega328-328P Datasheet complete”. Páginas 327

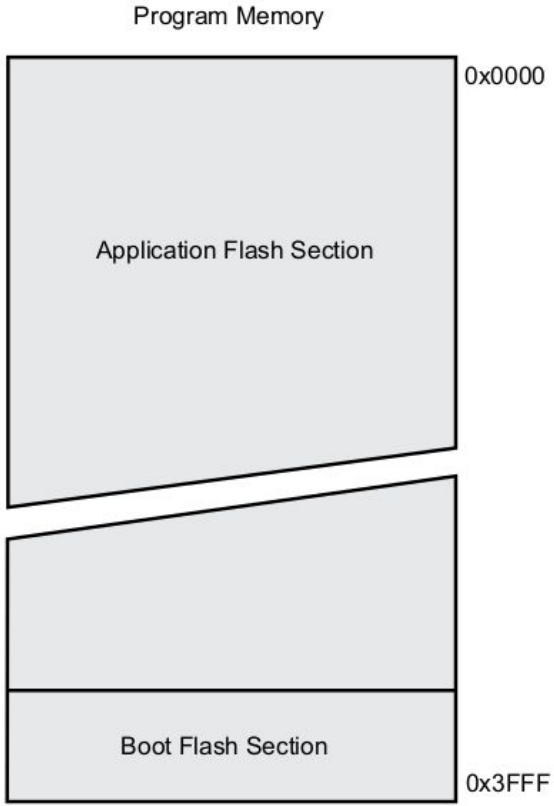
Organización de memoria de microcontroladores

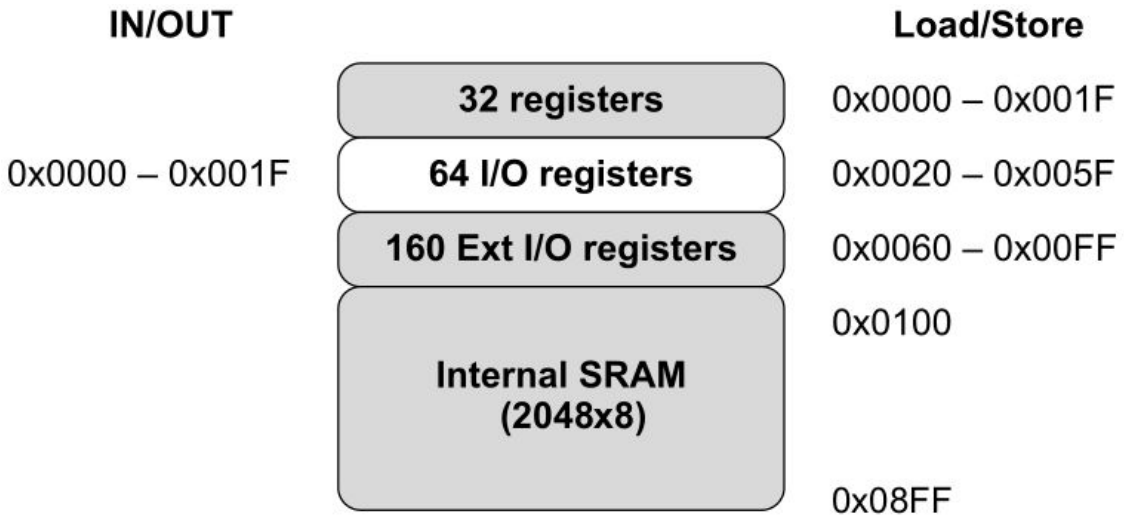
- Memorias no volátiles:
  - **Flash**: se borra y escribe eléctricamente de **varias posiciones** a la vez.
    - Usada como **Memoria de programa** o **Tablas** cuyo contenido no cambia.
    - Accedida desde
      - Boot loader (para cargar el programa)
      - Mecanismos de programación propios del microcontrolador (usualmente paralelo o serie).
      - Desde el programa en ejecución (no en todos los microcontroladores).
    - Límite típico de escrituras: 10 mil.

Organización de memoria de microcontroladores

- Memorias no volátiles:
  - **EEPROM**: se borra y escribe eléctricamente de **una posición** a la vez.
    - Usada como memoria de **datos** no volátil.
    - Accedida desde:
      - El programa en ejecución.
      - Mecanismos de programación propios del microcontrolador (usualmente paralelo o serie).
    - Límite típico de escrituras: 100 mil.
- Memorias volátiles:
  - SRAM o DRAM.
  - Accedida desde el programa en ejecución.
  - Puede haber un solo mapa para datos y I/O o varios mapas de memoria.

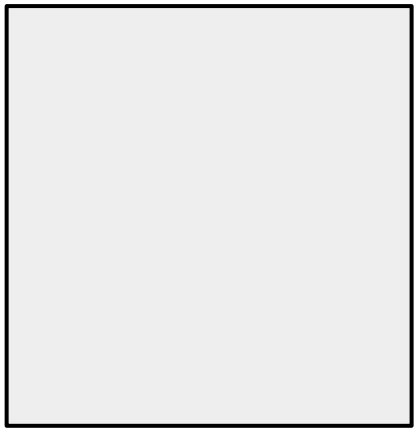
Organización de memoria de microcontroladores: ejemplo Atmel 328P





Figuras obtenidas de Atmel, “8-bit AVR Microcontrollers. ATmega328-328P Datasheet complete”. Página 34 y 35.

Organización de memoria de microcontroladores: ejemplo Atmel 328P
Memoria EEPROM



0

1023

- Tiempo de acceso mayor.
- Acceso controlado usualmente mediante interrupciones.
- La escritura requiere consumo de energía mayor (la fuente de alimentación debe poder proveerlo sin que Vcc disminuya).
- Es usual escribir y luego leer para verificar la correcta escritura.

Figuras obtenidas de Atmel, “8-bit AVR Microcontrollers. ATmega328-328P Datasheet complete”. Página 34 y 35.

Boot Loader

- Programa que permite escribir y leer la memoria flash de un microcontrolador, y por lo tanto cargar o leer el programa en el microcontrolador.
  - Ocupa lugar en la memoria flash.
  - Puede sobreescribirse o borrarse.

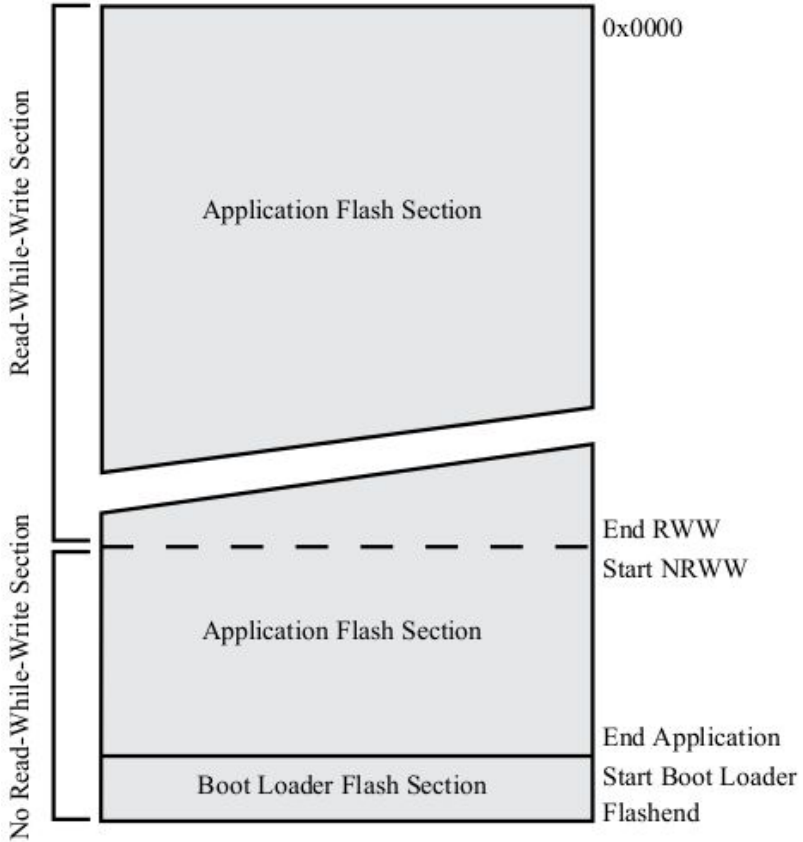
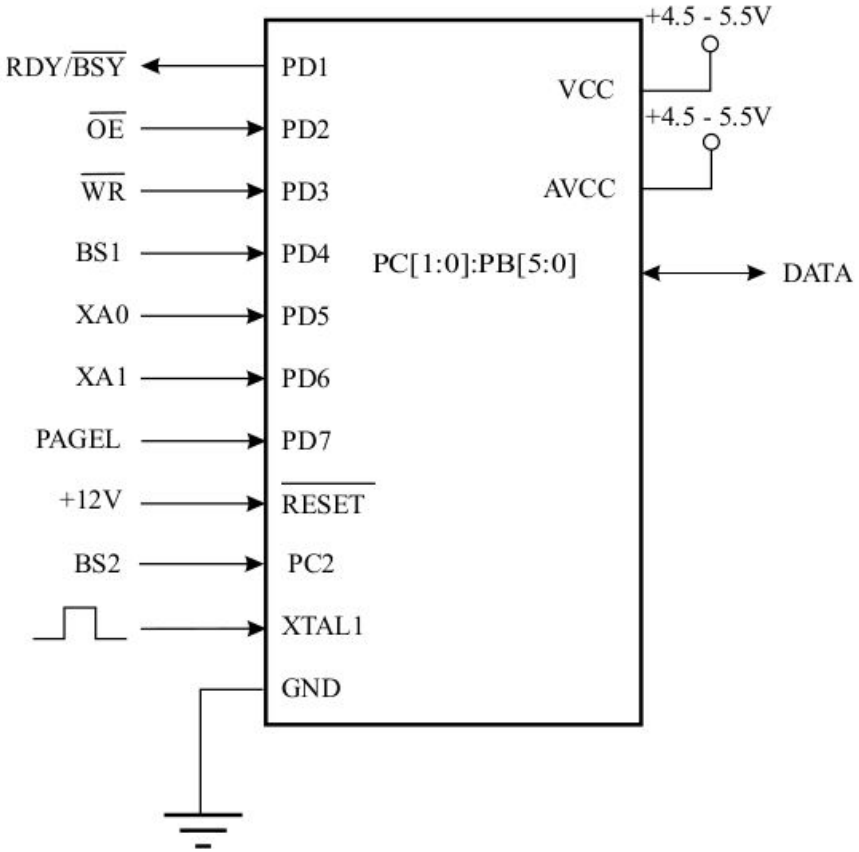


Figura obtenida de Atmel, “8-bit AVR Microcontrollers. ATmega328-328P Datasheet complete”. Página 333

Programación paralela de la memoria no volátil

- Se requiere seguir una secuencia de pasos (protocolo de escritura).
- Ejemplo Atmel 328P:
  - Aplicar 0 volt a pines especificados del microcontrolador, incluyendo Vcc.
  - Aplicar entre 4.5 y 5.5 volts a Vcc.
  - Esperar entre 20 y 60 µs y aplicar 11.5 y 12.5 volts al pin RESET.
  - Enviar comandos de programación.
    - A través del bus de datos (ver filmina siguiente).
  - Finalizar aplicando 0 volts al pin RESET.



Programación memoria FLASH

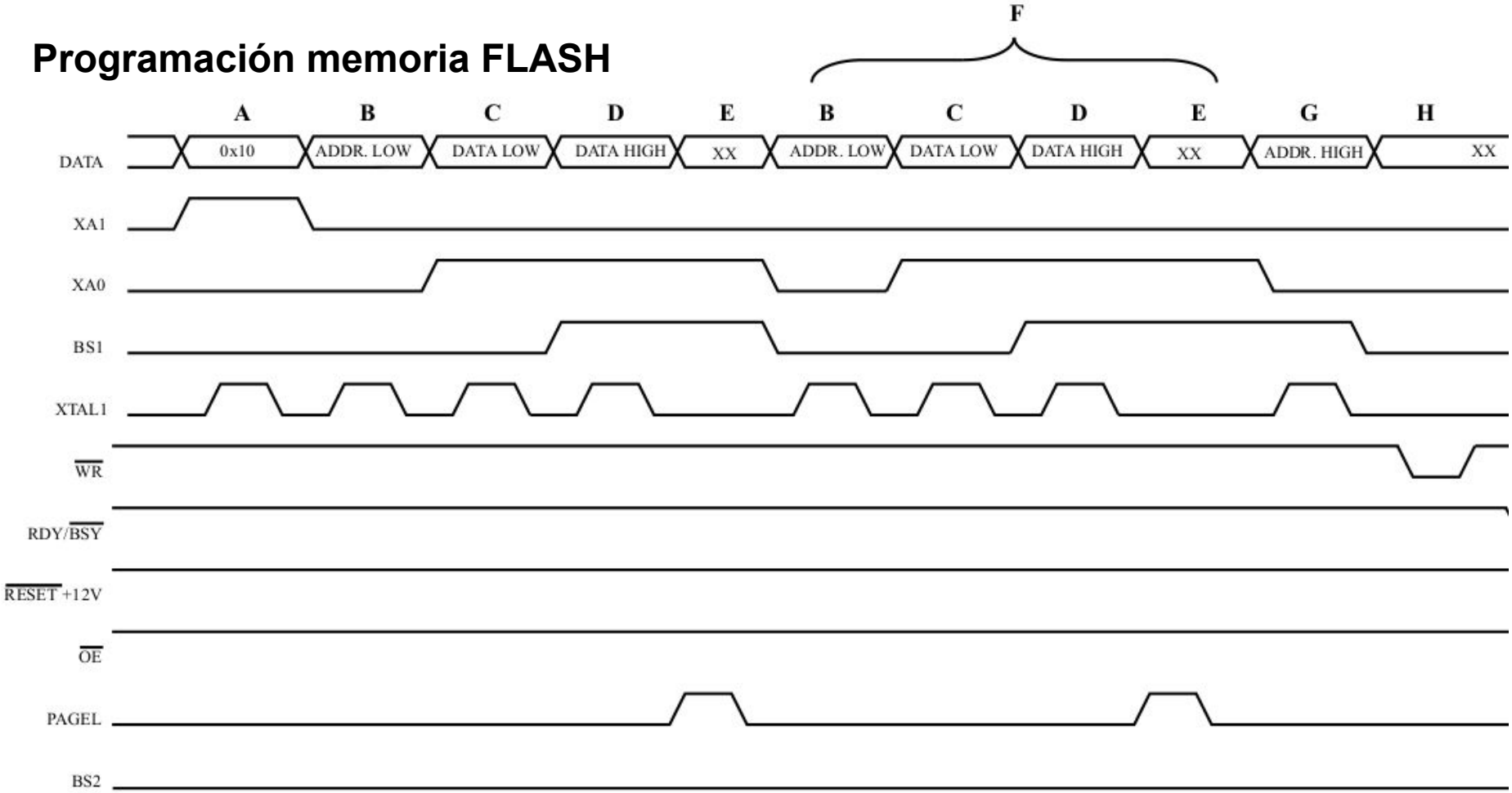


Figura obtenida de Atmel, “8-bit AVR Microcontrollers. ATmega328-328P Datasheet complete”. Página 356

Programación memoria EEPROM

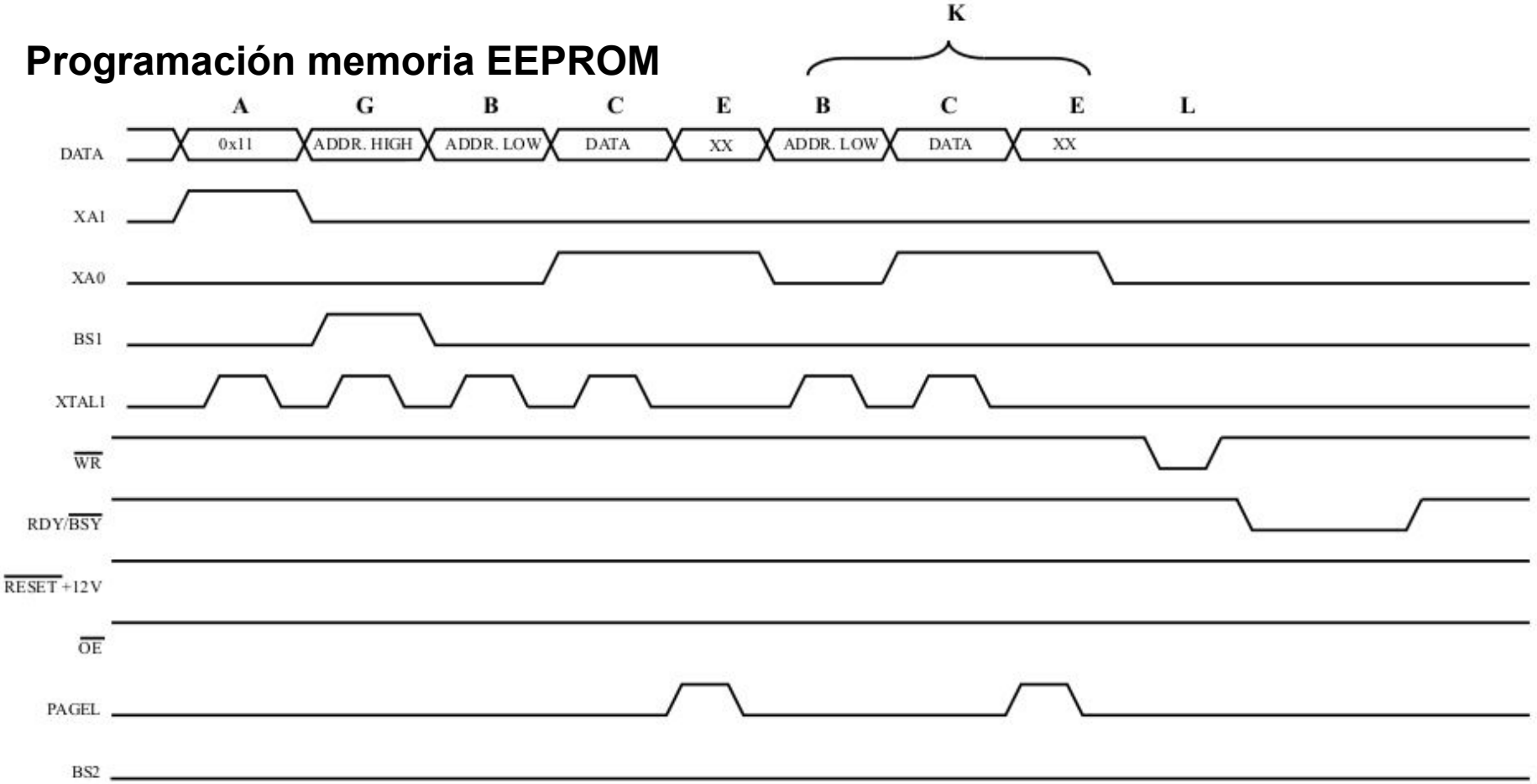
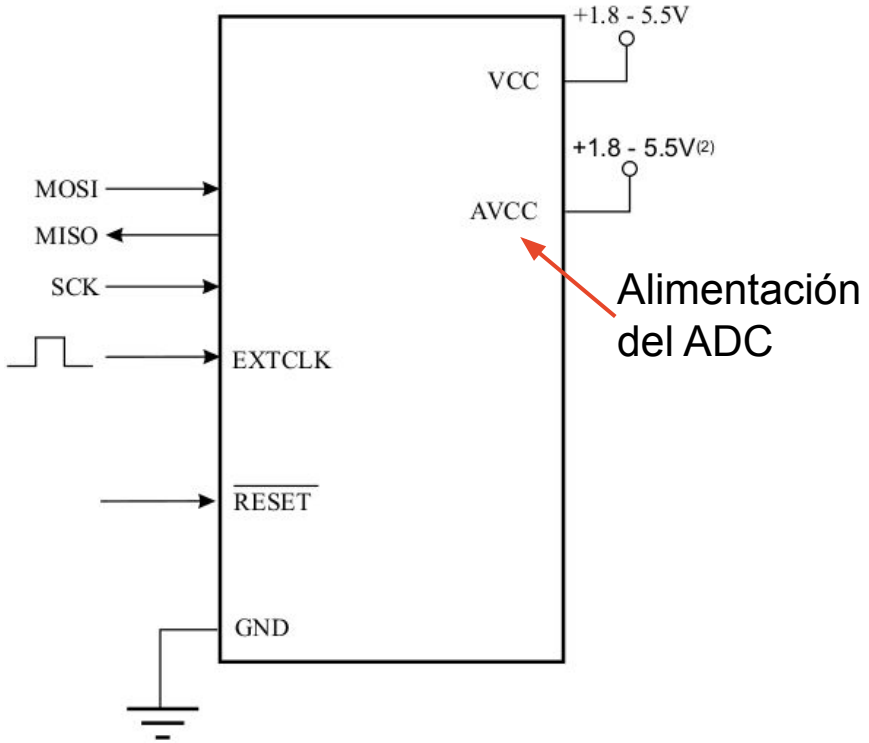


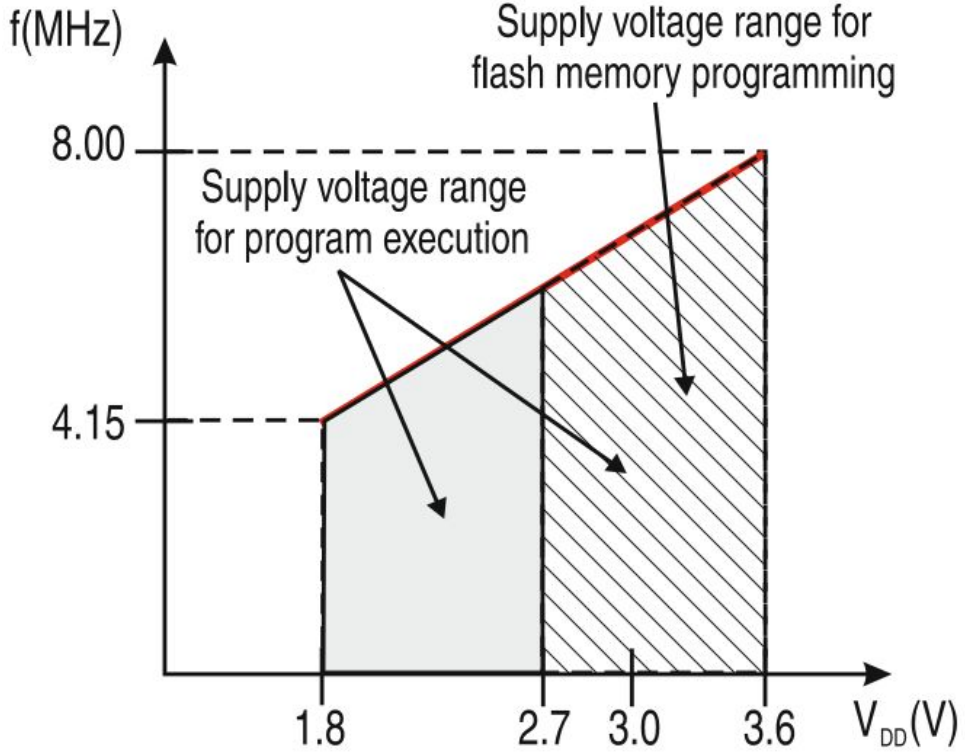
Figura obtenida de Atmel, “8-bit AVR Microcontrollers. ATmega328-328P Datasheet complete”. Página 356

Programación serie de la memoria no volátil

- Debe realizarse una secuencia de pasos.
- Los bits se comunican en serie.
- Ejemplo Atmel 328P:
  - Se utiliza el puerto SPI.
  - Se coloca RESET y AVCC a 0 volt.
  - Se envían instrucciones.



Memoria no volátil



Consumo energía y tensión alimentación durante la escritura de la memoria flash

Figura obtenida de Jiménez et. a. “Introduction to Embedded Systems Using Microcontrollers and the MSP430”. Página 253.

Bibliografía:

- Manuel Jiménez, Rogelio Palomera, Isidoro Couvertier, “Introduction to Embedded Systems Using Microcontrollers and the MSP430”. Springer. 2014.
- Muhammad Ali Mazidi, Sarmad Naimi, Sepehr Naimi. "The AVR microcontroller and embedded system: using Assembly and C". Pearson. 2011
- John Davies, "MSP430 Microcontroller Basics”. Elsevier. 2008.
- Microchip Technology Inc. PIC12F629/675 Data Sheet.
- Atmel, “8-bit AVR Microcontrollers. ATmega328-328P Datasheet complete”. 2016.
- Manuel Jiménez, Rogelio Palomera, Isidoro Couvertier, “Introduction to Embedded Systems Using Microcontrollers and the MSP430”. Springer. 2014.
- Peter Marwedel, "Embedded System Design. Embedded Systems, Foundations of Cyber-Physical Systems, and the Internet of Things". Springer. 2018.
- David Russel, "Introduction to Embedded System Using ANSI C and the

Bibliografía:

- Chaves Osorio, José & Quintero, Edwin & Cortes, Jimy. (2011). Generación de señales senoidales mediante PWM y filtros activos de segundo orden.
- Tanenbaum, Wetherall. "Computer Networks". 5° edición .Prentice Hall. 2011.
- Tom Igoe, Don Coleman, Brian Jepson, "Beginning NFC". O'Reilly. 2014
- LoRa Alliance. "A technical overview of LoRa and LoRaWAN". 2015.