



UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO



**FACULTAD
DE INGENIERÍA**

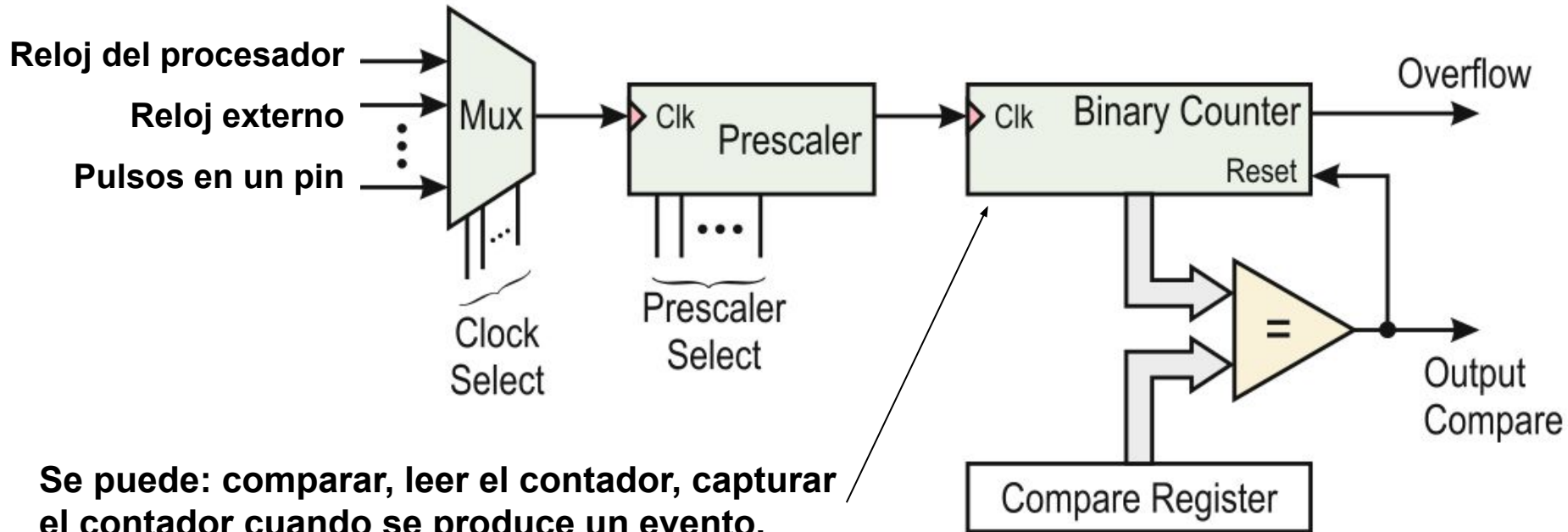
**Licenciatura en Ciencias de la
Computación**

Sistemas Embebidos

Unidad 4

Timers - Memorias no volátiles.

Timer: esquema general con registro comparador





Temporizadores (Timers)

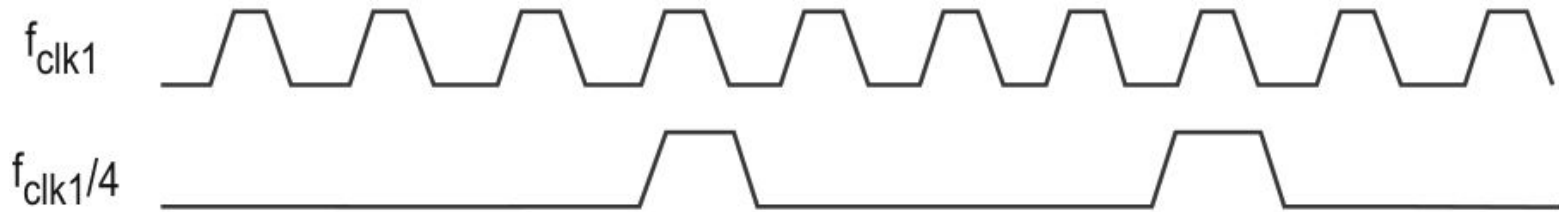
- **Funciones:**
 - Temporizadores (generar alguna señal a intervalos regulares de tiempo).
 - Contadores de eventos.
 - Relojes de tiempo real.
 - Generación de velocidad en baudios.
 - Modulación de ancho de pulso (PWM).
 - Mediciones de tiempo entre dos eventos.
 - Temporizadores watchdog.
 - Generar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



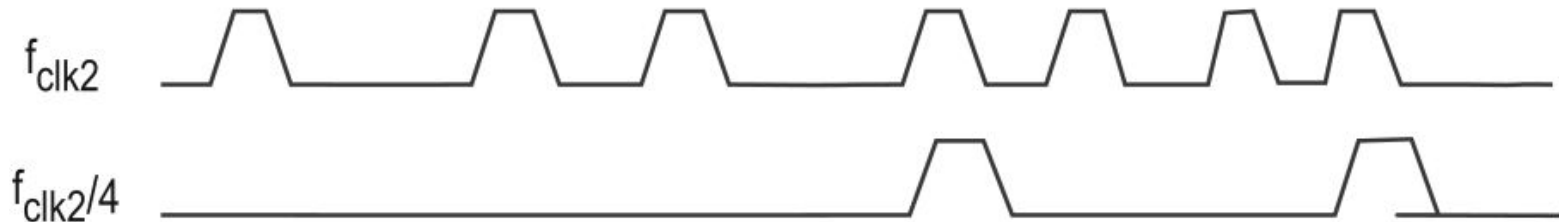
Timer: esquema general

- 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)



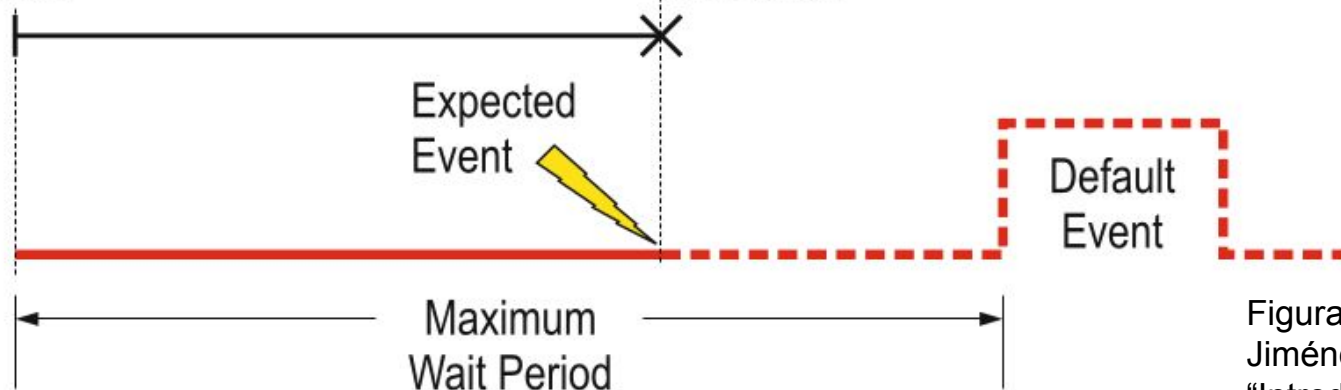
(b)

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).



WDT
Start

Timer
Expired

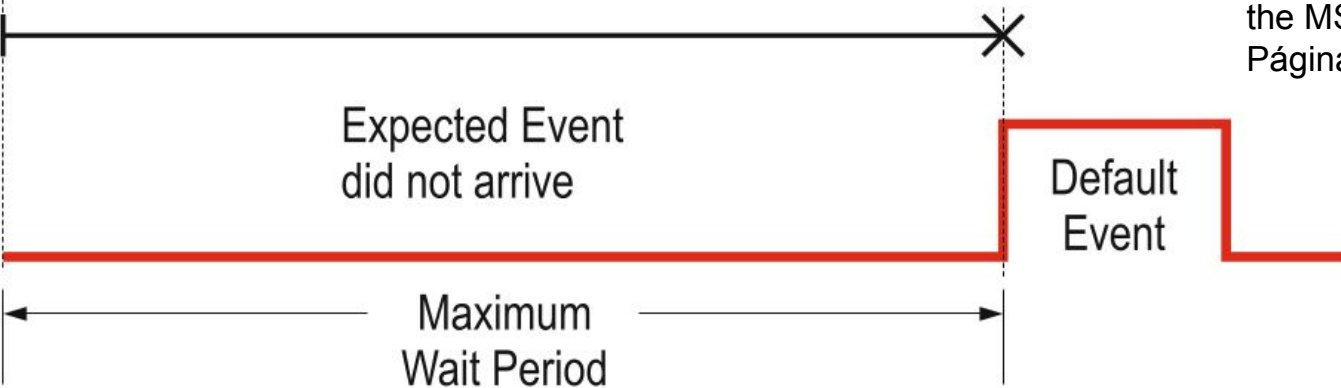


Figura obtenida de
Jiménez et. al.
“Introduction to
Embedded Systems
Using
Microcontrollers and
the MSP430”.
Páginas 336 y 337



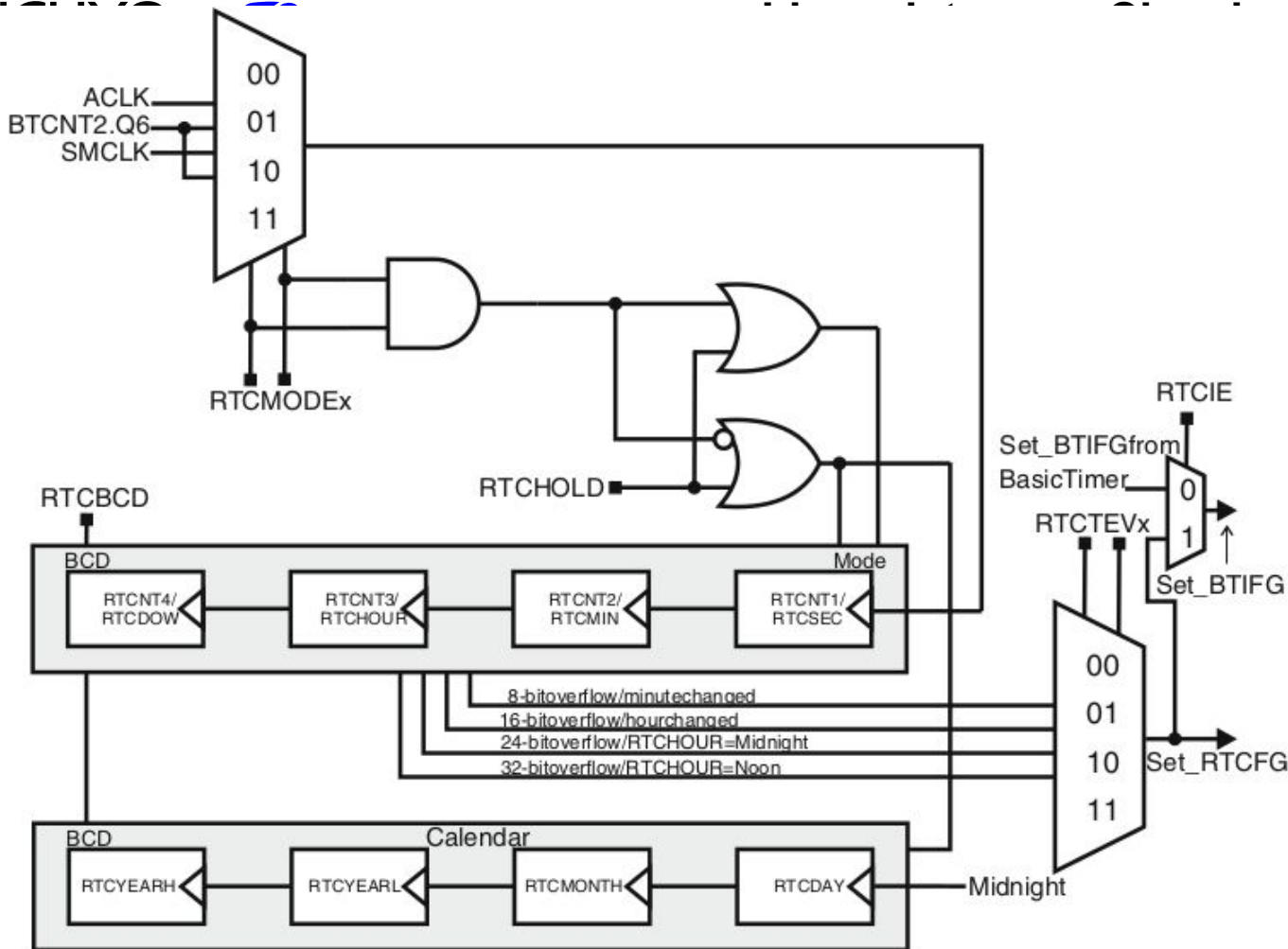
Real time Clock

- 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.)



UNI
NA

de la





Organización de memoria de microcontroladores

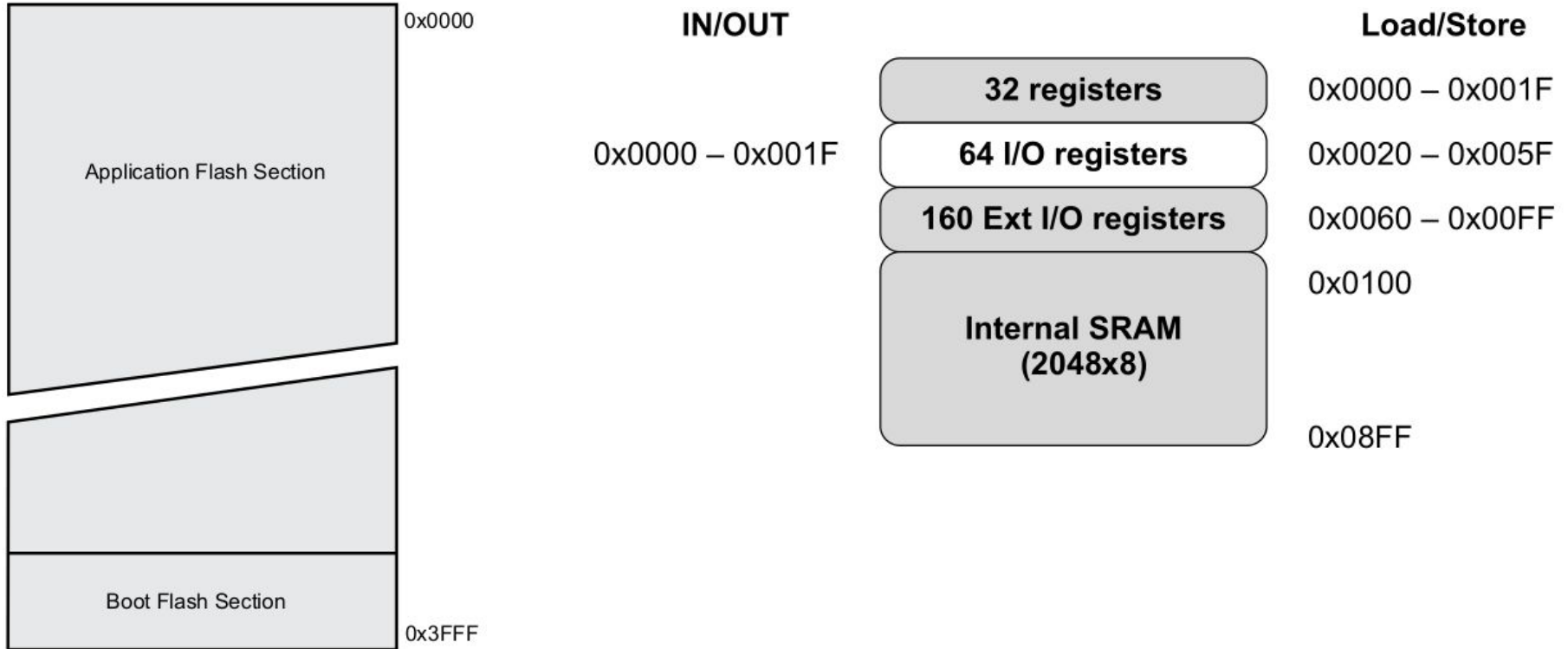
- **Memorias no volátiles:**
 - **Flash:** se borra y escribe eléctricamente de **varias posiciones** a la vez.
 - Usada como **Memoria de programa** o **Tablas** cuyo contenido no cambia.
 - Accedida desde
 - Boot loader (para cargar el programa)
 - Mecanismos de programación propios del microcontrolador (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

Program Memory



Organización de memoria de microcontroladores: ejemplo Atmel 328P

Memoria EEPROM



0

1023

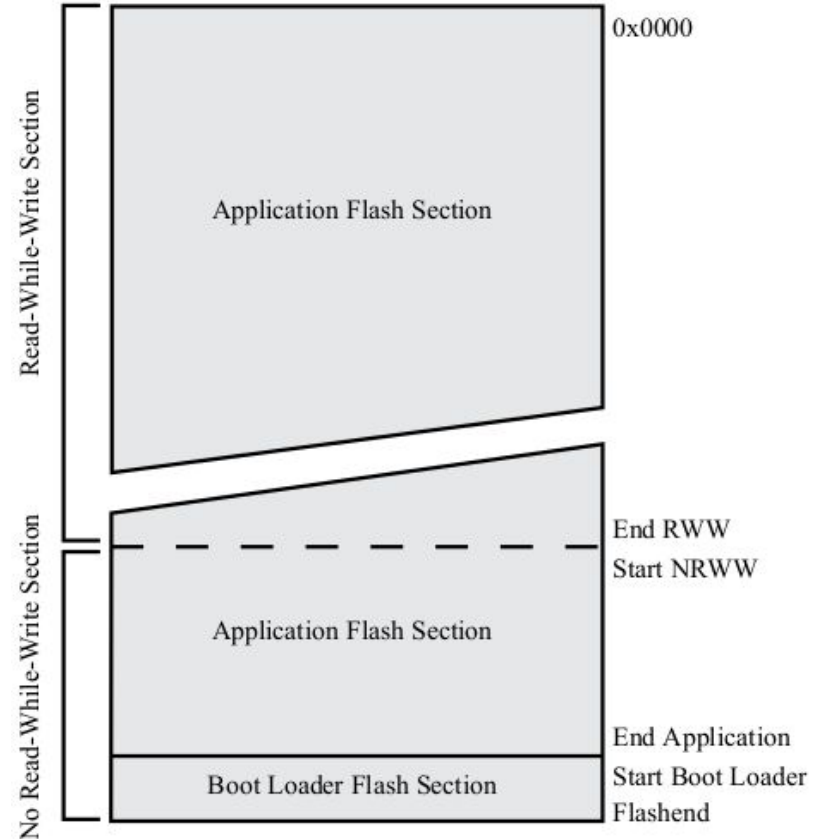
- Tiempo de acceso mayor.
- Acceso controlado usualmente mediante interrupciones.
- La escritura requiere consumo de energía mayor (la fuente de alimentación debe poder proveerlo sin que V_{cc} disminuya).
- Es usual escribir y luego leer para verificar la correcta escritura.



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 sobrescribirse o borrarse.

Licenciatura en Ciencias de la Computación



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.