

# Programación en Lenguaje Ensamblador

## Sistemas con Microprocesadores

[Ing. Esteban Volentini \(evolentini@herrera.unt.edu.ar\)](mailto:evolentini@herrera.unt.edu.ar)

<http://microprocesadores.unt.edu.ar/procesadores>

# Cronograma

Actividad	Inicio	Descripción	Fin
Presentación	19/08	Reglamento de la Materia	✓
Tema 1	19/08	Estructura de las computadoras	✓
Tema 2	26/08	Proyecto con un microcontrolador	✓
Tema 3	30/08	Descripción funcional de microprocesador	✓
Tema 4	13/09	Programación en lenguaje ensamblador	✓
Tema 5	25/09	Descripción general de un microcontrolador	←
Tema 6	27/09	Estructura general de microcontrolador	
Parcial	09/10	Primer examen parcial	
Tema 7	14/10	Sistema de Interrupciones	
Tema 8	21/10	Entradas y salidas digitales	
Tema 9	28/10	Entrada/salida con perifericos	
Tema 10	06/11	Temporizadores	
Proyectos	25/11	Seminarios de Proyectos	
Parcial	04/12	Segundo examen parcial	

# Bibliografía

---

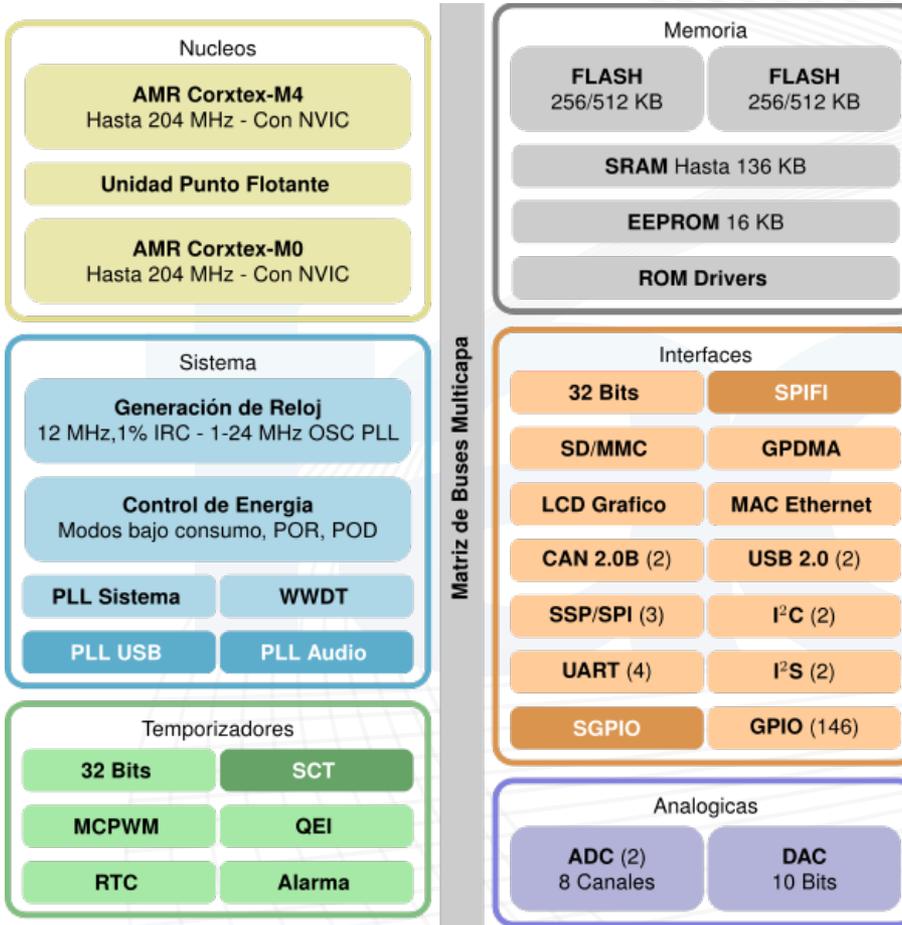
- ▶ **UM10503\_NXP\_LPC4337**
  - ▶ LPCxx User Manual – Capítulo 1.
- ▶ Daremos solo un panorama general.
- ▶ En temas que siguen veremos en detalle algunos periféricos.

# Descripción NXP LPC43xx

---

- ▶ Todos los microcontroladores de la familia NXP LPC43 están diseñados utilizando diversos módulos que se ocupan de tareas específicas.
  - ▶ Objetivo: diseño de aplicaciones con un mínimo de Hw adicional.
- ▶ Pasaremos revista a los módulos que componen el LPC43xx.
- ▶ Cada módulo puede configurarse modificando diversos registros de control y leerse mediante diversos registros de datos y Status.
  - ▶ Existe una configuración inicial que debe incluirse en la rutina de reset.

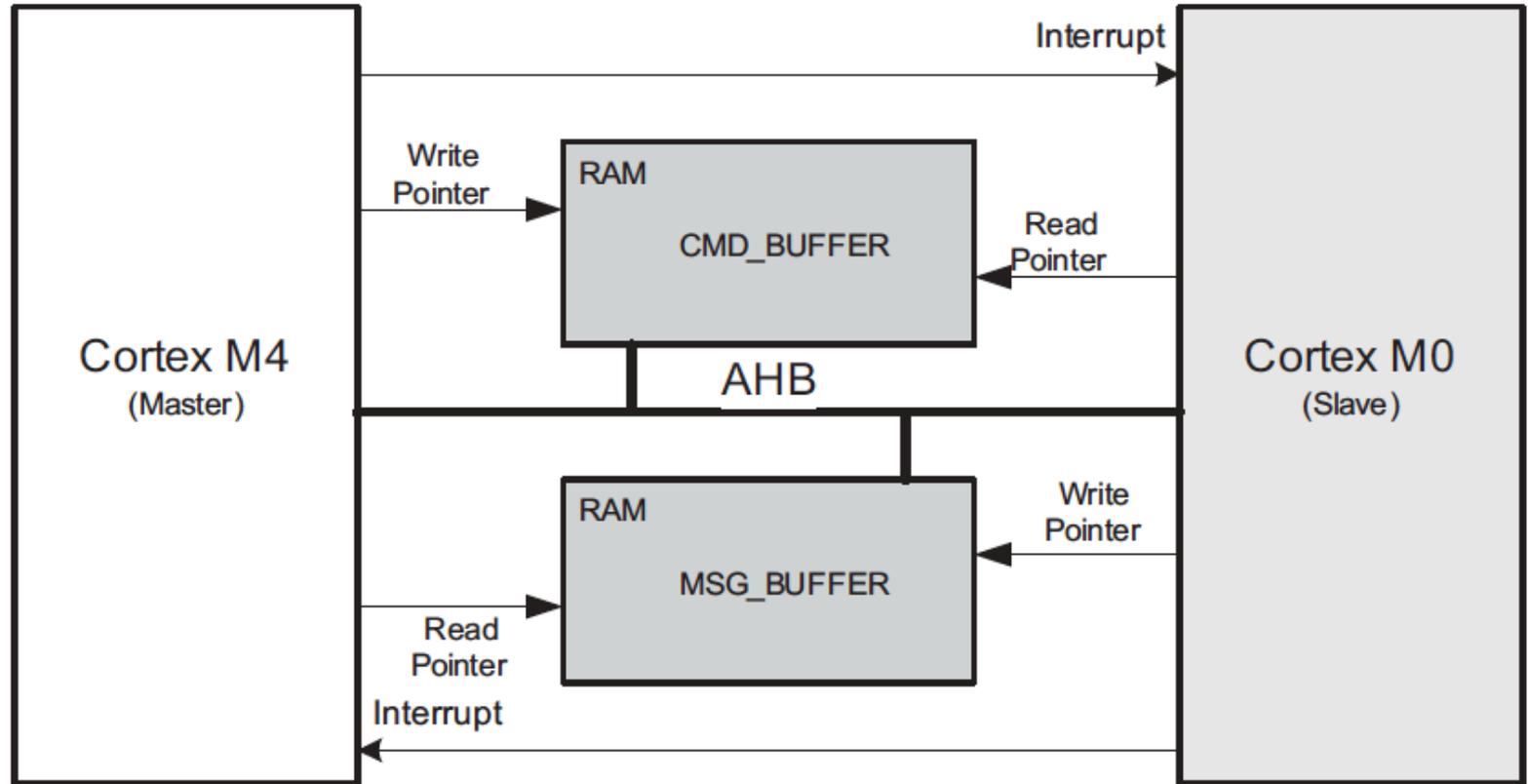
# Dual-Core Cortex-M4/M0



- ▶ Dual Core Cortex M4/M0 204 MHz
- ▶ Perifericos avanzados
  - ▶ 2 conexiones USB alta velocidad.
  - ▶ Conexión Ethernet.
  - ▶ Conexión Flash Externa como si fuera interna.
  - ▶ SCT, SGPIO, SPIFI

Part Number	Flash (kB)	SRAM (kB)	Ethernet MAC	HS USB	LCD	SD/ MMC
LPC4312	512**	104				
LPC4313	512	104				
LPC4315	768	136				
LPC4317	1024	136				
LPC4322	512**	104		1		
LPC4323	512	104		1		
LPC4325	768	136		1		
LPC4327	1024	136		1		
LPC4333	512	136	1	2		Y
LPC4337	1024	136	1	2		Y
LPC4353	512	136	1	2	Y	Y
LPC4357	1024	136	1	2	Y	Y

# Coordinación M4 – M0

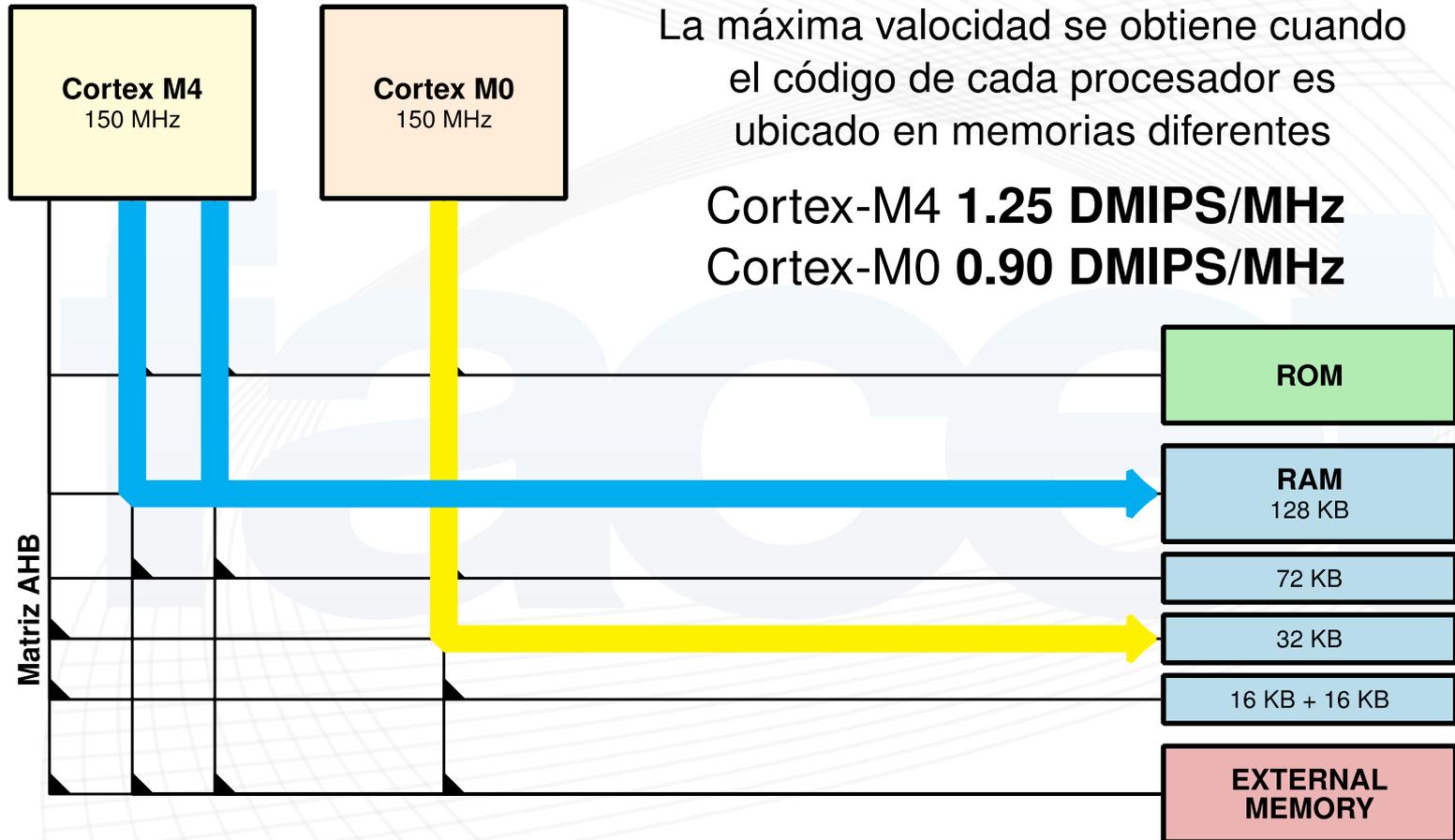


# Descripción: Cores

---

- ▶ Dos CPUs. asimétricos
  - ▶ M4 – Capacidad de Procesamiento, PDS, etc.
  - ▶ M0 – Para realizar tareas de E/S (IOP) y asegurar tiempo real.
- ▶ Bus: Matriz de Interconexión. Cada CPU puede manejar un bus en paralelo con el otro.

# Uso de Buses en Paralelo



# Assembler

---

- ▶ M0 – Thumb1 – 16 bits.
- ▶ M4 – Thumb2 – 16 y 32 bits.
  - ▶ División en Hw.
  - ▶ Soporte Procesamiento Digital de Señales.
- ▶ Thumb2 es compatible hacia abajo con Thumb1.
  - ▶ Códigos en Thumb1 se pueden ejecutar en M4

# Memoria

---

- ▶ Dos bancos de Memoria Flash y dos buses rápidos
  - ▶ En una el programa y en la otra datos fijos.
  - ▶ Se pueden leer en paralelo para el CPU.
- ▶ Hasta 1 MB de Flash en los dos bancos.
- ▶ Hasta 264 KB SRAM.
- ▶ Se puede colocar Flash Externa (serial) como parte del mapa de Memoria.
  - ▶ SPI o Quad SPI (4 conexiones seriales).

# Conversión A/D y D/A

---

- ▶ ADC (2)
  - ▶ Precisión 10 Bits.
  - ▶ Canales Multiplexados: hasta 22 entre los dos.
  - ▶ Permite el ingreso de varias señales analógicas.
  - ▶ Puede interrumpir con el fin de conversión o mostrar un status simplemente.
- ▶ DAC (1)
  - ▶ Resolución 10 bits.
  - ▶ Interrumpe al fin de conversión o muestra estado.

# Modos de bajo consumo

---

- ▶ Se ingresa mediante instrucciones
  - ▶ WFI, WFE.
  - ▶ Se configura en qué modo entran.
  - ▶ Sleep. Llega tensión pero no clock al core.
  - ▶ Cada core independientemente. Se conserva el estado.
- ▶ **Deep Sleep:** La RAM mantiene su estado.
  - ▶ Clock no alimenta cores ni periféricos.
  - ▶ Se mantiene el estado lógico de Cores y Periféricos.
- ▶ **Power Down:** Mantiene solo 8 KB SRAM
  - ▶ No conserva estados lógicos (no se alimenta al resto).
- ▶ **Deep Power Down:** No se guarda nada en SRAM.
  - ▶ Solo RTC funciona. Contiene 256 bytes para guardar lo fundamental.
  - ▶ Cuando se sale se ejecuta una rutina de booteo.

# Cómo salir de Bajo Consumo

---

- ▶ RTC – real time clock. Fija fecha y hora. Funciona permanentemente.
- ▶ Oscilador Interno 32KHz, alimenta un timer de alarma. Se debe configurar.
- ▶ Solo estos timers pueden sacar al CPU de modos de bajo consumo
- ▶ Mediante pedidos de interrupción o eventos que generan señales wake-up.

# Reset

---

- ▶ Se realiza desde FLASH
- ▶ La ROM incluye módulos para varios periféricos y también para escribir en FLASH y EEPROM.
- ▶ En Reset  $PC \leftarrow M(4)$ ,  $SP \leftarrow M(0)$
- ▶ Se puede configurar para iniciar
  - ▶ desde FLASH (externa) o USB (según estado de ciertos Pins).

# PLL – Phase Locked Loop

---

- ▶ Para generar señales de reloj estables.
  - ▶ A partir de cristal externo u oscilador interno RC.
- ▶ Se puede configurar la frecuencia ( $\leq 205\text{MHz}$ )
- ▶ Hay varias señales de reloj que se distribuyen en el CPU.
- ▶ Hay 3 PLLs
  - ▶ Audio – System – USB.
- ▶ A mayor frecuencia, mayor consumo.
- ▶ CPUs de alta performance permiten menor frecuencia para la misma tarea.
- ▶ También realizar tareas complejas a costa de mayor consumo.

# Watchdogs

---

- ▶ Son timers descendentes que al llegar a cero producen una acción determinada.
- ▶ Ej: un contador que usa un oscilador propio que no se detiene en modo bajo consumo.
  - ▶ Al llegar a 0 da una señal para despertar.
- ▶ Se configura la duración del mismo.
- ▶ En forma genérica se le llama watchdog (perro guardián).
  - ▶ “Duermo” pero el watchdog “me despertará”.

# WWDT (Windowed Watchdog Timer)

---

- ▶ Es un contador descendente. Configurable.
- ▶ Periódicamente el programa debe cargarlo.
  - ▶ Entre un mínimo y máximo de tiempo: ventana.
  - ▶ Si no lo hace ⇒ mal funcionamiento.
- ▶ Reset cuando llega a 0.
- ▶ Interrupción para avisar que está fuera de la ventana.
- ▶ El mínimo es para asegurar que no haya un mal funcionamiento, ej un lazo lo carga muy seguido.
- ▶ WWDT no cuenta en deep-sleep en adelante.
- ▶ En sleep, genera una interrupción para despertar al CPU.
- ▶ No cargarlo en interrupciones ¿por qué?
  - ▶ Interrupciones Cortas ¿por qué?
  - ▶ ¿Vale la pena usarlo siempre?

# Timers (cont)

---

- ▶ Varios tipos de timers.
- ▶ Para medir duración de señales.
- ▶ Para generar señales, de música, de control de motores, etc.
- ▶ Para tener fecha y hora siempre (RTC).
- ▶ Para medir tiempos, **como el caso de SysTick.**
- ▶ Watchdogs
  - ▶ Para despertar de bajo consumo
  - ▶ Para caso de errores.
- ▶ Veremos uno de los varios timers en detalle.

# Causas de Reset

---

- ▶ POR – Power On Reset.
- ▶ Pin externo /RESET – configurarlo.
- ▶ POD – detecta baja tensión y pide reset.
  - ▶ Mejor que no funcione antes de que funcione mal.
- ▶ WWDT – mal funcionamiento del Sw.
- ▶ La causa queda en un registro.
  - ▶ Se puede leer por programa de booteo y tomar acción adecuada.
  - ▶ Ej: display “batería baja”.

# Periféricos

---

- ▶ **GPIO.**
  - ▶ Puerto de E/S de propósito general.
  - ▶ Configurable, veremos en todo detalle.
  - ▶ Hasta 16 bits en paralelo.
  - ▶ Se puede elegir cuántos usar.
- ▶ **SGPIO.**
  - ▶ GPIO serial.
  - ▶ Varios puertos.
  - ▶ Se puede configurar el protocolo de comunicación.
- ▶ **Diversas interfases para comunicación.**
  - ▶ Se verán algunas en “Transmisión de Datos”.

# Interrupciones

---

- ▶ Internas de todos los módulos.
  - ▶ Para interactuar con CPU, debe configurarse.
- ▶ Externas, GPIO
  - ▶ Se pueden configurar hasta 8 interrupciones externas independientes.
    - ▶ Configurando 8 pins de GPUs para ello.
    - ▶ Veremos en detalle más adelante.
  - ▶ Hasta 2 interrupciones grupales
    - ▶ AND y OR de entradas seleccionadas de GPUs.
    - ▶ Entradas pueden negarse o no