

# Microprocesadores

Apuntes de clase

Javier Rodrigo López <sup>1</sup>

18 de febrero de 2021



<sup>1</sup>Correo electrónico: [javiolonchelo@gmail.com](mailto:javiolonchelo@gmail.com)



UNIVERSIDAD  
POLITÉCNICA  
DE MADRID



## Introducción

---

Imagen de la portada: *Dante y Virgilio en el infierno*, por William-Adolphe Bouguereau.

Contenidos

5 Temas

- Tema 0. Presentación (S1)
- Tema 1. Memorias semiconductoras (S1-S2 + S5)
- Tema 2. Microprocesadores (S2-S4)
- Tema 3. Micro (S6-S8)
- Resolución de problemas (S5, S15-S16)

Laboratorio - Aula D8211

Calendario

Profesorado Manuel César Rdoríguez Lacruz (coordinador) - Temas 0 a 3 Miguel Chavarrías Lapastora - Tema 4

Recursos

- The definitive guide to the ARM Cortex-M0 and Cortex-M0+ processors. Newnes
- Fast and effective embedded systems design. Applying the ARM mbed. Newnes.
- Embedded C Coding Standard. Barrgroup. **IMPORTANTE, TODO EL CÓDIGO TIENE QUE TENER ESTE ESTILO**

Excepto que las llaves de apertura van al final de la línea y la anchura de las indentaciones es de 2 espacios, en lugar de 4 como marca el libro.

Evaluación

Laboratorio (asistencia obligatoria)

$$L = 0.05P_1 + 0.125P_2 + 0.125P_3 + 0.2P_4 + 0.2L_1 + 0.3L_2(*)$$

Hay varias modalidades de examen de laboratorio. Si se aprueba el laboratorio, solo se realiza el examen de teoría.

Teoría

Calificación final Hay que sacar como mínimo un 4.5 en teoría y laboratorio para poder hacer media.

Requisitos

- Hay que saber manejar MUY BIEN el hexadecimal y el binario natural.
- Hay que saber hacer divisores resistivos y circuitos de polarización de diodos y transistores (bipolares y FET)

- Comprender los subsistemas combinacionales y secuenciales (incluyendo autómatas) y aplicar sus técnicas de análisis y síntesis.
- Codificar programas en lenguaje C/C++
- Aplicar las herramientas de *debugging* y demás

Asistencia obligatoria.

Materiales para el laboratorio

- Una o dos *protoboards*.
- Una placa STM Nucleo-l432kc. Hay que desoldar un par de jumpers. Buscar en la foto.
- Un modelo concreto de sensor ultrasónico.
- Cable de pares 1 metro (evitar los cables Dupont).
- 3 LED
- 3 micropulsadores
- 2 displays de 7 segmentos
- 1 LDR
- Para el resto de los componentes, mirar las listas de las prácticas, en Moodle.

#### Lista de tareas

---

1. Ir instalando los programas para las prácticas.
2. Descargar libros de bibliografía.
3. Comprar material.



# Índice general

Introducción . . . . .	2
<b>1. Memorias semiconductoras . . . . .</b>	<b>7</b>
1.1. Bancos de registros . . . . .	7
1.1.1. Estructura . . . . .	7
1.1.2. Chip select . . . . .	7
1.1.3. Doble puerto . . . . .	7
1.1.4. Bus bidireccional . . . . .	7
1.1.5. Output enable . . . . .	7
1.1.6. Capacidad y organización . . . . .	7
1.1.7. Símbolo . . . . .	8
1.1.8. Cronograma de funcionamiento . . . . .	8
1.2. Memorias semiconductoras . . . . .	8
1.2.1. Características . . . . .	8
1.2.2. Clasificación de los sistemas de almacenamiento . . . . .	8
1.2.3. Parámetros . . . . .	9
1.2.4. Cronogramas de funcionamiento . . . . .	9
1.2.5. Mapas de memoria . . . . .	9
1.3. Mapas de memoria . . . . .	10
<b>2. Microprocesadores . . . . .</b>	<b>11</b>
2.1. Concepto de algoritmo . . . . .	11
2.2. Sistemas secuenciales con memoria. Definición de microprocesador . . . . .	11
2.3. Elementos internos de un microprocesador . . . . .	11
2.4. Arquitectura de tres buses . . . . .	11
2.5. Ejemplos de codificación de instrucciones . . . . .	11
2.6. Evolución de los microprocesadores . . . . .	11
2.7. Modelos de programación y set de instrucciones . . . . .	11
2.8. Pila . . . . .	11
2.9. Característica de las arquitecturas . . . . .	11
2.10. Entorno de programación para sistemas empujados . . . . .	11
<b>3. Procesador ARM Cortex-M0 . . . . .</b>	<b>13</b>
3.1. Historia de ARM . . . . .	13
3.2. Arquitectura ARM Cortex-M0 . . . . .	13
3.2.1. Características principales de la arquitectura . . . . .	13
3.2.2. Organización de memoria . . . . .	13
3.2.3. Modelo de programación . . . . .	13
3.2.4. Set de instrucciones . . . . .	13
3.2.5. Reset del procesador . . . . .	13
3.2.6. Tamaños de datos . . . . .	13
3.3. Microcontroladores basados en arquitecturas ARM Cortex-M . . . . .	13
3.3.1. NXP LPC1768 . . . . .	13
3.3.2. STM ST32L432KC . . . . .	13
<b>4. Técnicas de I/O e interrupciones . . . . .</b>	<b>15</b>
4.1. Entrada/Salida . . . . .	16
4.2. GPIO . . . . .	16
4.3. Interrupciones . . . . .	16
4.3.1. Polling e interrupciones . . . . .	16
4.3.2. Esquemas hardware para la gestión de interrupciones . . . . .	16
4.3.3. Esquemas hardware para la gestión de interrupciones . . . . .	16

4.3.4. Conceptos de enmascaramiento, vector, prioridad, latencia, anidamiento y excepción . . . . .	16
4.3.5. Sleep . . . . .	16
4.3.6. Particularización para la arquitectura Cortex-M0 . . . . .	16
4.4. Temporizadores . . . . .	16
4.5. PWM . . . . .	16
4.6. ADC y DAC . . . . .	16
4.7. Sistemas controlados por eventos . . . . .	16
4.7.1. Concepto de sistema reactivo y de evento . . . . .	16
4.7.2. Máquinas de estados finitos controladas por eventos . . . . .	16
4.8. Comunicaciones serie asíncronas . . . . .	16
4.8.1. Concepto . . . . .	16
4.8.2. Parámetros y variantes . . . . .	16
4.8.3. Interfaz físico . . . . .	16
4.8.4. UART y transceiver . . . . .	16
4.8.5. Programación . . . . .	16
<b>5. Laboratorio</b>	<b>17</b>
5.1. Lenguaje de ensamble . . . . .	17
5.2. Entrada/Salida . . . . .	17
5.3. Temporizadores e interrupciones . . . . .	17
5.4. Diseño de aplicación de mediana complejidad . . . . .	17

# Capítulo 1

## Memorias semiconductoras

### 1.1 Bancos de registros

---

Un **registro** es un circuito secuencial paralelo-paralelo que permite almacenar un dato de varios bits.

Un **banco de registros** es una colección de registros. Se usan para almacenar varios datos de varios bits.

Al hablar de registros, es normal denominarlos **direcciones**. Entonces, en un banco de registros de 8 registros, cuando se quiera escribir un dato en el quinto registro se dirá que se quiere acceder a la dirección 5.

#### 1.1.1. Estructura

Vamos a ver la estructura con un ejemplo. Un registro de cuatro datos de 8 bits usando registros de 8 bits. Una salida de 8 bits. Por lo tanto, necesitaremos un multiplexor de 4 a 1 con su correspondiente entrada de selección **A\_RD** (dirección de lectura). Además, se necesita una entrada de habilitación **A\_WR** (dirección de escritura) que, usando un decodificador, permitirá la habilitación de los registros, teniendo este mismo otra entrada de habilitación denominada **WRn** (escritura, a nivel bajo).

#### 1.1.2. Chip select

Chip Select (**CSn**, a nivel bajo) es un enable del banco de registros. Se añade una puerta AND con entradas **CSn** y **WRn**, la salida será **WEn**, que es la entrada de habilitación del decodificador.

#### 1.1.3. Doble puerto

El **doble puerto** de lectura consiste en añadir otra salida, con su correspondiente multiplexor y entrada de selección de registro de lectura.

#### 1.1.4. Bus bidireccional

El bus bidireccional es un único bus que hace las funciones de bus de entrada y bus de salida. Se necesita el **buffer de tres estados** (falta describir este componente).

#### 1.1.5. Output enable

El output enable (**OEn**) es una entrada que proporciona permiso para obtener la lectura por el bus de datos.

#### 1.1.6. Capacidad y organización

En esta asignatura, los datos van a ser denominados **palabras**.

- $n \equiv$  número de registros
- $p \equiv$  anchura del bus de direcciones ( $n = 2^p$ )
- $q \equiv$  tamaño en bits de cada palabra
- Organización:  $n \times q$ . Ejemplo: 32x64

- Capacidad:  $n \cdot q$ . Ejemplo:  $32 \cdot 64 = 2048$  palabras

### 1.1.7. Símbolo

### 1.1.8. Cronograma de funcionamiento

FALTA AÑADIR IMÁGENES

## 1.2 Memorias semiconductoras

---

Son registros con algunas diferencias.

- No emplean flip-flops para almacenar la información. Permiten almacenar mucha más información que los bancos de registros.
- Muchas veces son asíncronas, careciendo de reloj y de puerto de inicialización síncrona.

Sin embargo, comparten bastantes características:

- Los conceptos de organización y capacidad son idénticos.
- Existen variantes con buses separados, doble puerto...

### Pasos a seguir para escribir una palabra en la memoria

---

1. Poner la dirección a la que quieres acceder.
2. Poner el dato que quieres escribir.
3. Habilitar la memoria.
4. Decirle que NO vas a leer. Esto es obligatorio.
5. Siempre tiene que haber un flanco de subida para escribir el dato, porque las memorias son asíncronas. La escritura sucede durante el flanco.

### 1.2.1. Características

#### Permanencia

---

Una vez haya sido almacenada la información, su contenido no se puede alterar (solo lectura/permanentes vs. lectura/escritura)

#### Volatilidad

---

Si una memoria es volátil, la información se pierde al retirar la alimentación.

#### Acceso

---

El tiempo de acceso a los datos es puede ser dependiente o independiente en la memoria o no. Son las memorias de acceso aleatorio y de acceso secuencial, respectivamente.

#### Dinámicas o estáticas

---

La información puede perderse al poco tiempo de ser almacenada y necesitas un refresco (*refresh*), o no. Son las memorias dinámicas y las memorias estáticas, respectivamente.

### 1.2.2. Clasificación de los sistemas de almacenamiento

- Magnéticos
  - Cintas
  - Discos
  - Núcleos de ferrita



- Ópticos
    - CD
    - DVD
  - Memorias semiconductoras
    - Acceso secuencial
      - Registros de desplazamiento
      - CCD (*Charge Coupled Devices*)
      - Pilas LIFO (Last-In / First-Out) y FIFO (First-In / First-Out)
    - Acceso aleatorio
      - Lectura y escritura
        - ◊ SRAM (Dynamic Random Access Memory) más rápida
        - ◊ DRAM (SDRAM (Synchronous Dynamic Random Access Memory)) más densa
        - ◊ Flash (no volátil)
      - Solo lectura
        - ◊ ROM
        - ◊ PROM
        - ◊ EPROM
        - ◊ E2PROM
- M: Memoria / O: *Read Only* / P: Programable / E(2): Erasable / E(1): Electrically

CAMBIAR ESTA LISTA POR UN ESQUEMA

### 1.2.3. Parámetros

---

Cantidad de información (bits) que puede almacenar.

Tiempo que transcurre desde que se pide un dato a la memoria hasta que esta lo entrega.

- **Tiempo de acceso ( $t_a$ )**
- **Capacidad de almacenamiento**
- **Coste por bit:** Precio de la memoria dividido entre su capacidad en bits

Estas tres características están relacionadas. A menor  $t_a$ , menor capacidad y mayor coste.

#### Tiempos importantes

---

Control por RW

### 1.2.4. Cronogramas de funcionamiento

### 1.2.5. Mapas de memoria

En un sistema puede haber diversos tipos de memoria.

Por ejemplo, un ordenador puede necesitar memoria no volátil de lectura y escritura (flash) para almacenar un programa, y a la vez necesitar una memoria volátil estática (SRAM) para acceso de datos a alta velocidad.

Un **mapa de memoria** es una representación esquemática de qué tipo (chips o módulos)

## 1.3 Mapas de memoria

---

baia la cabaia

### **Ampliación del tamaño de palabra**

---

En ocasiones, se puede desear una memoria de cierta longitud de palabra, pero se dispone de chips de palabra menor. En tal caso, la palabra deseada se divide en fragmentos del tamaño de palabra que tengan los chips disponibles.

## Capítulo 2

# Microprocesadores

Existen problemas que no pueden ser resueltos mediante sistemas combinacionales.

### 2.1 Concepto de algoritmo

---

### 2.2 Sistemas secuenciales con memoria. Definición de microprocesador

---

### 2.3 Elementos internos de un microprocesador

---

### 2.4 Arquitectura de tres buses

---

### 2.5 Ejemplos de codificación de instrucciones

---

### 2.6 Evolución de los microprocesadores

---

### 2.7 Modelos de programación y set de instrucciones

---

### 2.8 Pila

---

### 2.9 Característica de las arquitecturas

---

### 2.10 Entorno de programación para sistemas empujados

---

Javier Rodrigo López

## Capítulo 3

# Procesador ARM Cortex-M0

### 3.1 Historia de ARM

---

### 3.2 Arquitectura ARM Cortex-M0

---

3.2.1. Características principales de la arquitectura

3.2.2. Organización de memoria

3.2.3. Modelo de programación

3.2.4. Set de instrucciones

3.2.5. Reset del procesador

3.2.6. Tamaños de datos

### 3.3 Microcontroladores basados en arquitecturas ARM Cortex-M

---

3.3.1. NXP LPC1768

3.3.2. STM ST32L432KC

Javier Rodrigo López



## Capítulo 4

# Técnicas de I/O e interrupciones

### 4.1 Entrada/Salida

---

### 4.2 GPIO

---

### 4.3 Interrupciones

---

#### 4.3.1. Polling e interrupciones

#### 4.3.2. Esquemas hardware para la gestión de interrupciones

#### 4.3.3. Esquemas hardware para la gestión de interrupciones

#### 4.3.4. Conceptos de enmascaramiento, vector, prioridad, latencia, anidamiento y excepción

#### 4.3.5. Sleep

#### 4.3.6. Particularización para la arquitectura Cortex-M0

### 4.4 Temporizadores

---

### 4.5 PWM

---

### 4.6 ADC y DAC

---

### 4.7 Sistemas controlados por eventos

---

#### 4.7.1. Concepto de sistema reactivo y de evento

#### 4.7.2. Máquinas de estados finitos controladas por eventos

#### Eventos y mensajes

---

#### Estados y variables extendidas, guardas

---

#### Codificación en C

---

#### Ejemplo de aplicación completa

---

### 4.8 Comunicaciones serie asíncronas

---

#### 4.8.1. Concepto

#### 4.8.2. Parámetros y variantes

#### 4.8.3. Interfaz físico

#### 4.8.4. UART y transceiver

#### 4.8.5. Programación



## Capítulo 5

# Laboratorio

5.1 Lenguaje de ensamble

---

5.2 Entrada/Salida

---

5.3 Temporizadores e interrupciones

---

5.4 Diseño de aplicación de mediana complejidad

---