

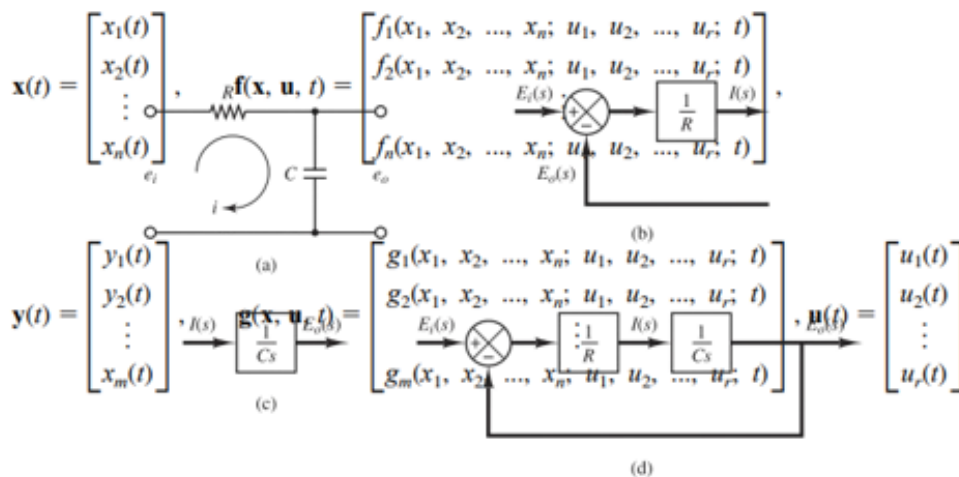


GUÍA-INSTRUMENTACIÓN II-FS543- FISMA

DEPÓSITO LEGAL: 202603221

ISSN: 3119-7957 (En línea)

Tutor: FIS. ING. JOSE LUIS HUAYANAY



Ayacucho-Perú
2026

Porque Jehová da la sabiduría, y de su boca viene el conocimiento y la inteligencia Prov.2:6 a Configuraci



Índice general

I		Parte Uno
1	Electrónica Digital y Sistemas de Conversión	11
1.1	Introducción	11
1.2	Señales Analógicas y Digitales	11
1.2.1	Definición de señal	11
1.2.2	Representación matemática	11
1.2.3	Ejemplo avanzado	12
1.3	Sistemas Numéricos	12
1.3.1	Sistema binario	12
1.3.2	Conversión decimal a binario	13
1.3.3	Ejemplo avanzado	13
1.4	Introducción a los Conversores Analógico-Digitales (ADC)	13
1.4.1	Concepto	13
1.4.2	Resolución	13
1.4.3	Ejemplo avanzado	13
1.5	Errores en la Conversión	14
1.5.1	Error de cuantización	14
1.5.2	Ejemplo avanzado	14
1.6	Aplicaciones Iniciales	14
1.7	Problemas Propuestos para casa	14
2	Compuertas Lógicas, Álgebra Booleana	17
2.1	Introducción	17

2.2	Compuertas Lógicas Básicas	17
2.2.1	Compuerta AND	17
2.2.2	Compuerta OR	17
2.2.3	Compuerta NOT	17
2.3	Compuertas Derivadas	18
2.3.1	Compuerta NAND	18
2.3.2	Compuerta NOR	18
2.3.3	Compuerta XOR	18
2.4	Álgebra Booleana	18
2.4.1	Postulados básicos	18
2.4.2	Leyes de De Morgan	18
2.5	Simplificación de Funciones Booleanas	19
2.5.1	Método algebraico	19
2.5.2	Mapas de Karnaugh	19
2.6	Diseño de Circuitos Combinacionales	19
2.6.1	Ejemplo avanzado: Sumador completo	19
2.6.2	Implementación con compuertas	19
2.7	Familia Lógica TTL	19
2.7.1	Características principales	19
2.7.2	Niveles lógicos típicos	19
2.7.3	Ejemplo: Circuito TTL 7400	19
2.8	Diseño con TTL	20
2.8.1	Ejemplo avanzado	20
2.9	Análisis de Retardos	20
2.9.1	Ejemplo	20
2.10	Aplicaciones	20
2.11	Problemas Propuestos para casa	20
3	Circuitos Secuenciales y Sistemas de Memoria	23
3.1	Introducción	23
3.2	Concepto de Sistema Secuencial	23
3.3	Latches	23
3.3.1	Latch SR	23
3.3.2	Problema de indeterminación	24
3.4	Flip-Flops	24
3.4.1	Flip-Flop SR con reloj	24
3.4.2	Flip-Flop D	24
3.4.3	Flip-Flop JK	24
3.4.4	Flip-Flop T	24
3.5	Análisis Temporal	24
3.5.1	Tiempo de propagación	24
3.5.2	Frecuencia máxima	24
3.6	Registros	24
3.6.1	Registro de desplazamiento	24
3.6.2	Tipos	24

3.7	Contadores	25
3.7.1	Contador asíncrono	25
3.7.2	Contador síncrono	25
3.7.3	Ejemplo: contador binario de 3 bits	25
3.8	Diseño de Máquina de Estados	25
3.8.1	Modelo de Moore	25
3.8.2	Modelo de Mealy	25
3.9	Ejemplo Avanzado: Detector de Secuencia	25
3.10	Aplicaciones	25
3.11	Problemas Propuestos para casa	25
4	Aritmética Digital y Sistemas de Procesamiento	29
4.1	Introducción	29
4.2	Representación de Números	29
4.2.1	Sistema binario	29
4.2.2	Complemento a dos	29
4.2.3	Ejemplo avanzado	29
4.3	Sumadores	30
4.3.1	Medio sumador	30
4.3.2	Sumador completo	30
4.3.3	Sumador en cascada	30
4.4	Sumador de alta velocidad	30
4.4.1	Carry Look-Ahead	30
4.5	Restadores	30
4.5.1	Restador completo	30
4.5.2	Restador mediante complemento a dos	30
4.6	Multiplicadores	30
4.6.1	Multiplicación binaria	30
4.6.2	Multiplicador combinacional	31
4.7	División Binaria	31
4.7.1	Algoritmo básico	31
4.8	Unidad Aritmético-Lógica (ALU)	31
4.8.1	Estructura	31
4.8.2	Modelo matemático	31
4.8.3	Ejemplo avanzado	31
4.9	Análisis de Retardos	31
4.10	Optimización	31
4.11	Aplicaciones	31
4.12	Controladores Programables y Sistemas de Control Digital	32
4.12.1	Introducción	32
4.12.2	Arquitectura de un Controlador Programable	32
4.12.3	Modelo funcional	32
4.12.4	Ciclo de escaneo (Scan Cycle)	32

4.13	Lógica de Control	32
4.13.1	Equivalencia con lógica digital	32
4.13.2	Ejemplo aplicado	33
4.14	Lenguajes de Programación en PLC	33
4.14.1	Diagrama Ladder	33
4.14.2	Bloques funcionales	33
4.15	Integración con Sistemas Digitales	33
4.15.1	Relación con ADC	33
4.15.2	Procesamiento	33
4.15.3	Control	33
4.16	Ejemplo Avanzado: Sistema de Control Industrial	33
4.17	Análisis de Tiempo en Controladores	34
4.17.1	Ejemplo	34
4.18	Modelos Matemáticos de Sistemas de Control	34
4.18.1	Introducción	34
4.19	Modelado en el Dominio del Tiempo	34
4.19.1	Ecuaciones diferenciales	34
4.19.2	Ejemplo: sistema de primer orden	34
4.19.3	Respuesta al escalón	34
4.20	Transformada de Laplace	35
4.20.1	Definición	35
4.20.2	Función de transferencia	35
4.20.3	Ejemplo	35
4.21	Modelos en Espacio de Estados	35
4.21.1	Forma general	35
4.21.2	Ejemplo	35
4.22	Sistemas Discretos	35
4.22.1	Ecuación en diferencias	35
4.22.2	Transformada Z	35
4.23	Estabilidad del Sistema	35
4.23.1	Criterio	35
4.24	Controladores	36
4.24.1	Control proporcional (P)	36
4.24.2	Control PI	36
4.24.3	Control PID	36
4.25	Ejemplo Aplicado	36
4.26	Integración con Sistemas Digitales	36
4.26.1	Modelo completo	36
4.27	Problemas Propuestos para casa	36
5	Integración y Proyecto Aplicado	39
5.1	Introducción	39
5.2	Arquitectura General del Sistema	39
5.2.1	Bloques funcionales	40

5.3	Modelo Matemático del Sistema	40
5.4	Proyecto Integrador: Sistema de Control Inteligente	40
5.4.1	Descripción	40
5.4.2	Etapas del sistema	40
5.5	Diseño Lógico	40
5.5.1	Condiciones de control	40
5.5.2	Implementación	41
5.6	Diseño Secuencial	41
5.6.1	Máquina de estados	41
5.6.2	Transiciones	41
5.7	Implementación en Controlador (PLC)	41
5.7.1	Lógica Ladder	41
5.7.2	Ciclo de operación	41
5.8	Simulación del Sistema	41
5.8.1	Pasos	41
5.9	Ejemplo Aplicado Avanzado	41
5.10	Optimización del Sistema	42
5.11	Aplicaciones	42
5.12	Aplicación Avanzada: Sistema de Control con Arduino y Robot Colaborativo	42
5.12.1	Introducción	42
5.13	Arquitectura del Sistema	42
5.13.1	Componentes	42
5.14	Modelo Matemático	42
5.14.1	Conversión ADC	42
5.14.2	Control ON/OFF con histéresis	43
5.15	Implementación en Arduino	43
5.15.1	Código básico	43
5.16	Extensión: Control PID en Arduino	43
5.16.1	Modelo discreto	43
5.16.2	Código simplificado	44
5.17	Aplicación en Robot Colaborativo	44
5.17.1	Descripción	44
5.17.2	Modelo del sistema	44
5.18	Ejemplo: Control de Posición	44
5.19	Código simplificado (Servo)	45
5.20	Seguridad en Robots Colaborativos	45
5.21	Conclusión	45
5.22	Problemas Propuestos para casa	45
	Bibliografía	49
	Books	49
5.23	Bibliografía Recomendada	49

- 3.6 Registros
- 3.7 Contadores
- 3.8 Diseño de Máquina de Estados
- 3.9 Ejemplo Avanzado: Detector de Secuencia
- 3.10 Aplicaciones
- 3.11 Problemas Propuestos para casa

4 Aritmética Digital y Sistemas de Procesamiento **Parte Uno** 29

- 4.1 Introducción
- 4.2 Representación de Números
- 4.3 Sumadores
- 4.4 Sumador de alta velocidad
- 4.5 Restadores
- 4.6 Multiplicadores
- 4.7 División Binaria
- 4.8 Unidad Aritmético-Lógica (ALU)
- 4.9 Análisis de Retardos
- 4.10 Optimización
- 4.11 Aplicaciones
- 4.12 Controladores Programables y Sistemas de Control Digital
- 4.13 Lógica de Control
- 4.14 Lenguajes de Programación en PLC
- 4.15 Integración con Sistemas Digitales
- 4.16 Ejemplo Avanzado: Sistema de Control Industrial
- 4.17 Análisis de Tiempo en Controladores
- 4.18 Modelos Matemáticos de Sistemas de Control
- 4.19 Modelado en el Dominio del Tiempo
- 4.20 Transformada de Laplace
- 4.21 Modelos en Espacio de Estados
- 4.22 Sistemas Discretos
- 4.23 Estabilidad del Sistema
- 4.24 Controladores
- 4.25 Ejemplo Aplicado
- 4.26 Integración con Sistemas Digitales
- 4.27 Problemas Propuestos para casa

5 Integración y Proyecto Aplicado **39**

- 5.1 Introducción
- 5.2 Arquitectura General del Sistema
- 5.3 Modelo Matemático del Sistema
- 5.4 Proyecto Integrador: Sistema de Control Inteligente
- 5.5 Diseño Lógico
- 5.6 Diseño Secuencial
- 5.7 Implementación en Controlador (PLC)
- 5.8 Simulación del Sistema
- 5.9 Ejemplo Aplicado Avanzado
- 5.10 Optimización del Sistema
- 5.11 Aplicaciones
- 5.12 Aplicación Avanzada: Sistema de Control con Arduino y Robot Colaborativo
- 5.13 Arquitectura del Sistema
- 5.14 Modelo Matemático
- 5.15 Implementación en Arduino
- 5.16 Extensión: Control PID en Arduino
- 5.17 Aplicación en Robot Colaborativo
- 5.18 Ejemplo: Control de Posición
- 5.19 Código simplificado (Servo)
- 5.20 Seguridad en Robots Colaborativos
- 5.21 Conclusión
- 5.22 Problemas Propuestos para casa

Bibliografía **49**

- Books
- 5.23 Bibliografía Recomendada



1. Electrónica Digital y Sistemas de Conversión

1.1 Introducción

La presente guía de Instrumentación II se desarrolla a nombre de la Universidad Nacional de San Cristóbal de Huamanga, dirigida a estudiantes del área de Ciencias Físico-Matemáticas. Su propósito es incentivar el desarrollo de conocimientos teóricos y prácticos en instrumentación electrónica, así como promover la aplicación de tecnologías actuales en sistemas de medición, adquisición de datos y control.

El contenido está orientado a servir como una guía de aprendizaje de nivel avanzado, integrando conceptos fundamentales y aplicaciones modernas alineadas con las exigencias de la ingeniería contemporánea. Asimismo, se busca fortalecer las capacidades analíticas y experimentales del estudiante, fomentando el uso de herramientas tecnológicas y metodologías que permitan abordar problemas reales en distintos campos de la ciencia y la ingeniería.

La electrónica digital constituye la base de los sistemas modernos de control, procesamiento de datos y automatización industrial. Su evolución ha permitido el desarrollo de sistemas embebidos, controladores programables y arquitecturas computacionales altamente eficientes. En este capítulo se establecen los fundamentos necesarios para comprender el funcionamiento de los sistemas digitales, incluyendo la representación de señales, lógica binaria y los principios básicos de conversión analógico-digital.

1.2 Señales Analógicas y Digitales

1.2.1 Definición de señal

Una señal es una magnitud física que varía en el tiempo y transporta información. Estas pueden clasificarse en:

- **Señales analógicas:** continuas en el tiempo y amplitud.
- **Señales digitales:** discretas en el tiempo y amplitud.

1.2.2 Representación matemática

Una señal analógica se expresa como:

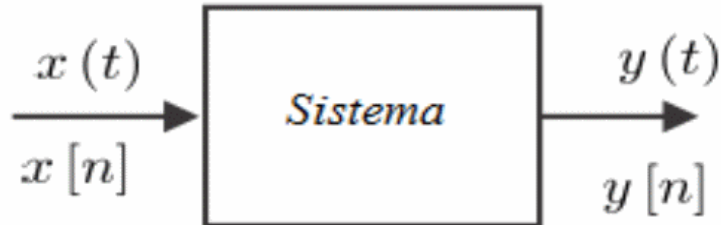


Figura 1.1.1: *electrónica digital constituye la base de los sistemas modernos*

$$x(t) \in \mathbb{R} \quad (1.2.1)$$

Mientras que una señal digital puede representarse como:

$$x[n] \in \{0, 1\} \quad (1.2.2)$$

1.2.3 Ejemplo avanzado

Considere una señal analógica sinusoidal:

$$x(t) = 5 \sin(2\pi 1000t) \quad (1.2.3)$$

Al ser muestreada a una frecuencia de 8 kHz:

$$x[n] = 5 \sin\left(2\pi 1000 \frac{n}{8000}\right) \quad (1.2.4)$$

Este proceso es la base de la digitalización de señales en sistemas ADC.

1.3 Sistemas Numéricos

1.3.1 Sistema binario

El sistema binario utiliza dos estados:

$$0 \rightarrow \text{Bajo}, \quad 1 \rightarrow \text{Alto} \quad (1.3.1)$$

1.3.2 Conversión decimal a binario

Ejemplo:

$$25_{10} = 11001_2 \quad (1.3.2)$$

1.3.3 Ejemplo avanzado

Conversión de número fraccionario:

$$0,625_{10} = 0,101_2 \quad (1.3.3)$$

Proceso:

$$0,625 \times 2 = 1,25 \rightarrow 1 \quad (1.3.4)$$

$$0,25 \times 2 = 0,5 \rightarrow 0 \quad (1.3.5)$$

$$0,5 \times 2 = 1,0 \rightarrow 1 \quad (1.3.6)$$

1.4 Introducción a los Conversores Analógico-Digitales (ADC)**1.4.1 Concepto**

Un ADC convierte una señal analógica en una representación digital discreta:

$$V_{in} \rightarrow Codigo\ digital \quad (1.4.1)$$

1.4.2 ResoluciónLa resolución de un ADC de n bits está dada por:

$$\Delta = \frac{V_{ref}}{2^n} \quad (1.4.2)$$

1.4.3 Ejemplo avanzadoPara un ADC de 10 bits con $V_{ref} = 5V$:

$$\Delta = \frac{5}{1024} \approx 4,88\text{mV} \quad (1.4.3)$$

Si la entrada es:

$$V_{in} = 2,5V \quad (1.4.4)$$

Entonces el valor digital es:

$$D = \frac{2,5}{4,88 \times 10^{-3}} \approx 512 \quad (1.4.5)$$

1.5 Errores en la Conversión

1.5.1 Error de cuantización

Se define como la diferencia entre el valor real y el valor digital:

$$e_q = V_{in} - V_{quantizado} \quad (1.5.1)$$

1.5.2 Ejemplo avanzado

Para un ADC de 3 bits:

$$V_{ref} = 8V \Rightarrow \Delta = 1V \quad (1.5.2)$$

Si:

$$V_{in} = 3,7V \quad (1.5.3)$$

Entonces:

$$V_{quantizado} = 4V \quad (1.5.4)$$

$$e_q = 3,7 - 4 = -0,3V \quad (1.5.5)$$

1.6 Aplicaciones Iniciales

- Sistemas de adquisición de datos
- Sensores industriales
- Sistemas de control automático
- Instrumentación electrónica

1.7 Problemas Propuestos para casa

Los siguientes preguntas propuestas como tareas, cada estudiante, debe entregar realizado a mano en hoja cuadriculada y si realiza la simulaciones ya sea en (Scilab, MATLAB o equivalente). Deberá dibujar a mano y visible.

1. Análisis de aliasing y reconstrucción de señal

Una señal está definida por:

$$x(t) = 10 \cos(2\pi 2000t) + 5 \sin(2\pi 3500t)$$

- Determine la frecuencia mínima de muestreo.
 - Analice el aliasing si se muestrea a 4 kHz.
 - Obtenga las frecuencias alias resultantes.
 - Proponga un método de reconstrucción y justifique matemáticamente.
- ### 2. Diseño completo de un sistema de muestreo
- Diseñe un sistema de adquisición para una señal biomédica con componentes hasta 1.2 kHz:
- Determine la frecuencia de muestreo adecuada.
 - Diseñe un filtro anti-aliasing (modelo matemático).

- Justifique el orden del filtro.
- Evalúe el error si el filtro no es ideal.

3. Conversión binaria avanzada

Convierta el número:

$$458,9375_{10}$$

a binario, hexadecimal y representación en complemento a dos de 16 bits.

- Analice el error de truncamiento.
- Discuta implicancias en sistemas digitales reales.

4. Modelado de cuantización

Para un ADC de 5 bits y $V_{ref} = 10V$:

- Determine niveles de cuantización.
- Modele la señal cuantizada para una entrada senoidal.
- Grafique el error de cuantización.
- Analice su distribución estadística.

5. Diseño de ADC de precisión

Diseñe un ADC para un sistema industrial con:

$$\text{Rango} = 0 - 12V, \quad \text{Resolución} < 2mV$$

- Determine el número de bits requerido.
- Analice el error máximo.
- Evalúe impacto del ruido.
- Proponga arquitectura ADC adecuada.

6. Simulación de error de cuantización

- Modele en Scilab una señal senoidal de 2 kHz.
- Simule su digitalización con ADC de 6, 8 y 10 bits.
- Compare errores.
- Analice resultados.

7. Análisis SNR en ADC

- Derive la expresión teórica del SNR en función de bits.
- Valide mediante simulación.
- Compare con resultados experimentales.

8. Diseño de sistema de adquisición multicanal

- Diseñe sistema con 4 sensores analógicos.
- Incluya multiplexación.
- Analice tiempos de muestreo.
- Evalúe error global.

9. Optimización de resolución

- Compare ADC de 8, 10 y 12 bits.
- Analice resolución vs costo.
- Proponga solución óptima para sistema embebido.

10. Error total en ADC real

Considere:

$$\text{Error total} = \text{Error cuantización} + \text{Error offset} + \text{Error ganancia}$$

- Modele matemáticamente cada componente.

- Evalúe impacto total.
 - Proponga método de compensación.
11. **Aliasing en señales no sinusoidales**
- Analice una señal cuadrada.
 - Determine componentes espectrales.
 - Evalúe aliasing.
12. **Diseño de filtro anti-aliasing**
- Diseñe filtro pasa-bajos.
 - Determine frecuencia de corte.
 - Justifique tipo de filtro (Butterworth, Chebyshev).
13. **Modelado matemático completo del ADC**
- Desarrolle ecuación general del proceso.
 - Incluya muestreo y cuantización.
 - Analice limitaciones.
14. **Análisis de precisión en sistemas reales**
- Evalúe influencia de temperatura.
 - Analice ruido eléctrico.
 - Proponga mejoras.
15. **Diseño de sistema de medición industrial**
- Diseñe sistema para sensor de presión.
 - Incluya ADC.
 - Evalúe errores.
16. **Comparación de arquitecturas ADC**
- Compare: Flash, SAR, Sigma-Delta.
 - Analice velocidad, precisión y costo.
 - Proponga aplicación ideal para cada uno.
17. **Análisis espectral de señal digitalizada**
- Realice FFT de señal muestreada.
 - Compare con señal original.
 - Analice distorsión.
18. **Diseño de sistema de adquisición con microcontrolador**
- Integre ADC interno.
 - Analice limitaciones.
 - Proponga mejoras.
19. **Simulación de sistema completo**
- Modele señal analógica.
 - Digitalice.
 - Reconstruya señal.
 - Compare resultados.
20. **Investigación aplicada**
- Analice un ADC comercial real.
 - Evalúe especificaciones.
 - Compare con modelo teórico.



2. Compuertas Lógicas, Álgebra Booleana

2.1 Introducción

Las compuertas lógicas constituyen la base fundamental de los sistemas digitales. A partir de ellas es posible construir circuitos combinacionales y secuenciales que permiten realizar operaciones lógicas, aritméticas y de control. En este capítulo se desarrollan los principios de las compuertas lógicas, el álgebra booleana, y su aplicación en el diseño de circuitos digitales, incluyendo características de la familia lógica TTL.

2.2 Compuertas Lógicas Básicas

2.2.1 Compuerta AND

La compuerta AND realiza el producto lógico:

$$Y = A \cdot B \tag{2.2.1}$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

2.2.2 Compuerta OR

$$Y = A + B \tag{2.2.2}$$

2.2.3 Compuerta NOT

$$Y = \bar{A} \tag{2.2.3}$$

FUNCIONES LÓGICAS BÁSICAS

NOMBRE	AND - Y	OR - O	XOR O-exclusiva	NOT Inversor	NAND	NOR																																																																																	
SÍMBOLO																																																																																							
SÍMBOLO																																																																																							
TABLA DE VERDAD	<table border="1"> <tr><td>a</td><td>b</td><td>z</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	a	b	z	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <tr><td>a</td><td>b</td><td>z</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	a	b	z	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <tr><td>a</td><td>b</td><td>z</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	a	b	z	0	0	0	0	1	1	1	0	1	1	1	0	<table border="1"> <tr><td>a</td><td>z</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	a	z	0	1	1	0	<table border="1"> <tr><td>a</td><td>b</td><td>z</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	a	b	z	0	0	1	0	1	1	1	0	1	1	1	0	<table border="1"> <tr><td>a</td><td>b</td><td>z</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	a	b	z	0	0	1	0	1	0	1	0	0	1	1	0
a	b	z																																																																																					
0	0	0																																																																																					
0	1	0																																																																																					
1	0	0																																																																																					
1	1	1																																																																																					
a	b	z																																																																																					
0	0	0																																																																																					
0	1	1																																																																																					
1	0	1																																																																																					
1	1	1																																																																																					
a	b	z																																																																																					
0	0	0																																																																																					
0	1	1																																																																																					
1	0	1																																																																																					
1	1	0																																																																																					
a	z																																																																																						
0	1																																																																																						
1	0																																																																																						
a	b	z																																																																																					
0	0	1																																																																																					
0	1	1																																																																																					
1	0	1																																																																																					
1	1	0																																																																																					
a	b	z																																																																																					
0	0	1																																																																																					
0	1	0																																																																																					
1	0	0																																																																																					
1	1	0																																																																																					
EQUIVALENTE EN CONTACTOS																																																																																							
AXIOMA	$z = a \cdot b$	$z = a + b$	$z = \bar{a} \cdot b + a \cdot \bar{b}$	$z = \bar{a}$	$z = \overline{a \cdot b}$	$z = \overline{a + b}$																																																																																	

Figura 2.1.1: Compuertas lógicas

2.3 Compuertas Derivadas

2.3.1 Compuerta NAND

$$Y = \overline{A \cdot B} \quad (2.3.1)$$

2.3.2 Compuerta NOR

$$Y = \overline{A + B} \quad (2.3.2)$$

2.3.3 Compuerta XOR

$$Y = A \oplus B = \bar{A}B + A\bar{B} \quad (2.3.3)$$

2.4 Álgebra Booleana

2.4.1 Postulados básicos

$$A + 0 = A \quad (2.4.1)$$

$$A \cdot 1 = A \quad (2.4.2)$$

$$A + A = A \quad (2.4.3)$$

$$A \cdot A = A \quad (2.4.4)$$

2.4.2 Leyes de De Morgan

$$\overline{A \cdot B} = \bar{A} + \bar{B} \quad (2.4.5)$$

$$\overline{A + B} = \bar{A} \cdot \bar{B} \quad (2.4.6)$$

2.5 Simplificación de Funciones Booleanas

2.5.1 Método algebraico

Ejemplo:

$$F = AB + A\bar{B} \quad (2.5.1)$$

Factorizando:

$$F = A(B + \bar{B}) = A \quad (2.5.2)$$

2.5.2 Mapas de Karnaugh

Ejemplo de simplificación de una función de 3 variables:

$$F(A, B, C) = \sum(1, 3, 5, 7) \quad (2.5.3)$$

Resultado simplificado:

$$F = C \quad (2.5.4)$$

2.6 Diseño de Circuitos Combinacionales

2.6.1 Ejemplo avanzado: Sumador completo

Entradas: A, B, C_{in}

$$S = A \oplus B \oplus C_{in} \quad (2.6.1)$$

$$C_{out} = AB + C_{in}(A \oplus B) \quad (2.6.2)$$

2.6.2 Implementación con compuertas

- Uso de compuertas XOR para suma
- Uso de AND y OR para acarreo

2.7 Familia Lógica TTL

2.7.1 Características principales

- Tecnología basada en transistores bipolares
- Alta velocidad de conmutación
- Consumo moderado de potencia

2.7.2 Niveles lógicos típicos

- Nivel bajo (0): $0 - 0,8V$
- Nivel alto (1): $2V - 5V$

2.7.3 Ejemplo: Circuito TTL 7400

El circuito 7400 contiene cuatro compuertas NAND.

2.8 Diseño con TTL

2.8.1 Ejemplo avanzado

Diseñar la función:

$$F(A, B, C) = \overline{(A + B)C} \quad (2.8.1)$$

Aplicando De Morgan:

$$F = \overline{A + B} + \overline{C} \quad (2.8.2)$$

Implementación:

- NOR para $(A + B)$
- NOT para C
- OR final

2.9 Análisis de Retardos

El tiempo de propagación es:

$$t_p = t_{PLH} + t_{PHL} \quad (2.9.1)$$

2.9.1 Ejemplo

Para tres compuertas en cascada:

$$t_{total} = t_1 + t_2 + t_3 \quad (2.9.2)$$

2.10 Aplicaciones

- Sistemas de control digital
- Microprocesadores
- Electrónica industrial
- Sistemas embebidos

2.11 Problemas Propuestos para casa

Los siguientes preguntas propuestas como tareas, cada estudiante, debe entregar realizado a mano en hoja cuadriculada y si realiza la simulaciones ya sea en (Scilab, MATLAB o equivalente). Deberá dibujar a mano y visible.

1. Diseño lógico completo desde especificación

Diseñe un sistema digital de 3 entradas que active una salida cuando al menos dos entradas estén en nivel alto.

- Construya la tabla de verdad.
- Obtenga la función booleana.
- Simplifique usando álgebra booleana y Karnaugh.
- Implemente con compuertas TTL.

2. Implementación usando solo NAND

Implemente la siguiente función únicamente con compuertas NAND:

$$F(A,B,C) = AB + \bar{A}C$$

- Transforme la función.
- Diseñe el circuito.
- Evalúe el número de compuertas necesarias.

3. Análisis de circuito lógico complejo

Dado un circuito con múltiples compuertas:

- Determine la función lógica equivalente.
- Simplifique.
- Compare el circuito original vs optimizado.

4. Diseño de sistema de alarma

Diseñe un sistema donde:

- Se activa si una puerta está abierta y el sensor de movimiento detecta presencia.
- Incluya condición de override manual.
- Modele la función lógica.
- Implemente con TTL.

5. Simplificación avanzada

Simplifique:

$$F = \bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC$$

- Use álgebra booleana.
- Verifique con mapa de Karnaugh.

6. Diseño de sumador completo optimizado

- Diseñe un sumador completo.
- Minimice el número de compuertas.
- Compare implementación XOR vs básica.

7. Análisis de retardos

- Diseñe un circuito con 4 niveles lógicos.
- Calcule retardo total.
- Evalúe impacto en frecuencia máxima.

8. Diseño con NOR exclusivamente

Implemente:

$$F = (A + B)(\bar{C} + D)$$

usando solo compuertas NOR.

9. Sistema de control industrial

Diseñe un sistema que controle un motor basado en:

- Sensor de temperatura
- Sensor de presión
- Botón de emergencia

Incluya lógica de seguridad.

10. Comparación TTL vs CMOS

- Analice diferencias eléctricas.

- Compare consumo y velocidad.
 - Proponga aplicación ideal para cada uno.
11. **Diseño de decodificador**
- Diseñe un decodificador 3 a 8.
 - Implemente con compuertas básicas.
 - Analice su uso en memoria.
12. **Multiplexor lógico**
- Diseñe un MUX 4:1.
 - Obtenga función booleana.
 - Implemente con compuertas.
13. **Análisis de hazard**
- Diseñe un circuito con hazard.
 - Identifique tipo de hazard.
 - Proponga solución.
14. **Optimización de circuitos**
- Compare dos implementaciones de la misma función.
 - Evalúe costo, velocidad y complejidad.
15. **Diseño de sistema de votación**
- Diseñe sistema de mayoría para 5 entradas.
 - Simplifique lógica.
 - Implemente.
16. **Análisis de potencia en TTL**
- Estime consumo total de un circuito.
 - Analice disipación térmica.
 - Proponga mejoras.
17. **Diseño de circuito aritmético**
- Diseñe un medio sumador y restador.
 - Integre ambos en un solo circuito.
18. **Simulación de circuito lógico**
- Modele un circuito en software.
 - Compare resultados teóricos vs simulados.
19. **Diseño jerárquico**
- Diseñe un sistema complejo usando módulos.
 - Integre subcircuitos.
20. **Investigación aplicada**
- Analice un circuito TTL real (ej: 7400, 7486).
 - Evalúe funcionamiento.
 - Compare con modelo teórico.



3. Circuitos Secuenciales y Sistemas de Memoria

3.1 Introducción

A diferencia de los circuitos combinatoriales, los circuitos secuenciales poseen memoria, es decir, su salida depende no solo de las entradas actuales, sino también de estados anteriores. Estos sistemas son fundamentales en aplicaciones como controladores programables, sistemas embebidos y procesamiento digital.

3.2 Concepto de Sistema Secuencial

Un sistema secuencial puede describirse como:

$$Y(t) = f(X(t), Estado(t)) \quad (3.2.1)$$

$$Estado(t + 1) = g(X(t), Estado(t)) \quad (3.2.2)$$

3.3 Latches

3.3.1 Latch SR

Implementado con compuertas NOR:

- S = Set
- R = Reset

S	R	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Indeterminado

3.3.2 Problema de indeterminación

La condición $S = R = 1$ genera un estado inválido, lo cual debe evitarse en diseño.

3.4 Flip-Flops

3.4.1 Flip-Flop SR con reloj

Introduce señal de control (clock):

$$Q(t+1) = S + \bar{R}Q(t) \quad (3.4.1)$$

3.4.2 Flip-Flop D

$$Q(t+1) = D \quad (3.4.2)$$

3.4.3 Flip-Flop JK

$$Q(t+1) = J\bar{Q}(t) + \bar{K}Q(t) \quad (3.4.3)$$

3.4.4 Flip-Flop T

$$Q(t+1) = T \oplus Q(t) \quad (3.4.4)$$

3.5 Análisis Temporal

3.5.1 Tiempo de propagación

$$t_p = t_{setup} + t_{hold} \quad (3.5.1)$$

3.5.2 Frecuencia máxima

$$f_{max} = \frac{1}{t_{total}} \quad (3.5.2)$$

3.6 Registros

3.6.1 Registro de desplazamiento

Permite almacenar y desplazar datos:

$$Q_{i+1} = Q_i \quad (3.6.1)$$

3.6.2 Tipos

- Serial-in Serial-out (SISO)
- Serial-in Parallel-out (SIPO)
- Parallel-in Serial-out (PISO)
- Parallel-in Parallel-out (PIPO)

3.7 Contadores

3.7.1 Contador asíncrono

- Cambios propagados en cascada
- Más lento

3.7.2 Contador síncrono

- Controlado por un mismo reloj
- Mayor velocidad

3.7.3 Ejemplo: contador binario de 3 bits

$$0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow 7 \quad (3.7.1)$$

3.8 Diseño de Máquina de Estados

3.8.1 Modelo de Moore

$$Y(t) = f(\text{Estado}(t)) \quad (3.8.1)$$

3.8.2 Modelo de Mealy

$$Y(t) = f(\text{Estado}(t), X(t)) \quad (3.8.2)$$

3.9 Ejemplo Avanzado: Detector de Secuencia

Diseñar un sistema que detecte la secuencia 101:

- Definir estados
- Construir diagrama de estados
- Obtener ecuaciones
- Implementar con flip-flops

3.10 Aplicaciones

- Memorias digitales
- Sistemas de control
- PLC
- Microprocesadores

3.11 Problemas Propuestos para casa

Los siguientes preguntas propuestas como tareas, cada estudiante, debe entregar realizado a mano en hoja cuadriculada y si realiza la simulaciones ya sea en (Scilab, MATLAB o equivalente). Deberá dibujar a mano y visible.

1. Diseño completo de latch SR con restricciones reales

- Diseñe un latch SR usando compuertas NOR.
- Analice el estado prohibido.

- Proponga una solución para evitar la indeterminación.
 - Simule el comportamiento temporal.
2. **Análisis de Flip-Flop D en sistema real**
 - Modele un Flip-Flop D con retardo.
 - Analice efectos de tiempo de setup y hold.
 - Determine condiciones de fallo.
 3. **Conversión entre Flip-Flops**
 - Diseñe un Flip-Flop JK usando Flip-Flop D.
 - Obtenga ecuaciones de conversión.
 - Verifique funcionamiento.
 4. **Diseño de registro de desplazamiento**
 - Diseñe un registro SIPO de 4 bits.
 - Modele su comportamiento temporal.
 - Analice aplicaciones.
 5. **Diseño de contador asíncrono**
 - Diseñe un contador binario de 4 bits.
 - Analice retardo acumulado.
 - Determine frecuencia máxima.
 6. **Diseño de contador síncrono optimizado**
 - Diseñe un contador de 4 bits síncrono.
 - Compare con el asíncrono.
 - Analice ventajas.
 7. **Diseño de contador módulo N**
 - Diseñe un contador módulo 10.
 - Determine lógica de reinicio.
 - Implemente con flip-flops.
 8. **Análisis de frecuencia máxima**
 - Dado un circuito secuencial, calcule frecuencia máxima.
 - Incluya retardos de compuertas.
 - Analice limitaciones.
 9. **Diseño de detector de secuencia**
 - Detectar secuencia 1101.
 - Diseñar máquina de estados (Moore y Mealy).
 - Comparar resultados.
 10. **Análisis de hazard temporal**
 - Identifique hazard en circuito secuencial.
 - Clasifique tipo.
 - Proponga solución.
 11. **Diseño de registro universal**
 - Diseñe un registro con modos:
 - Carga paralela
 - Desplazamiento izquierda/derecha
 - Modele su funcionamiento.
 12. **Contador reversible**
 - Diseñe un contador ascendente/descendente.
 - Incluya control de dirección.
 - Implemente con flip-flops.
 13. **Diseño de divisor de frecuencia**
 - Diseñe un circuito que divida la frecuencia por 8.

- Analice comportamiento temporal.
- 14. **Análisis de sincronización**
 - Evalúe un sistema con múltiples flip-flops.
 - Analice problemas de sincronización.
 - Proponga mejoras.
- 15. **Diseño de memoria básica**
 - Diseñe una celda de memoria de 1 bit.
 - Escale a 4 bits.
 - Analice funcionamiento.
- 16. **Máquina de estados compleja**
 - Diseñe una FSM para control de semáforo.
 - Modele estados.
 - Implemente con flip-flops.
- 17. **Simulación completa**
 - Modele un sistema secuencial completo.
 - Simule entradas y salidas.
 - Compare resultados teóricos.
- 18. **Análisis de consumo dinámico**
 - Evalúe consumo en flip-flops.
 - Analice impacto de frecuencia.
 - Proponga optimización.
- 19. **Diseño de sistema digital aplicado**
 - Diseñe sistema de control de acceso.
 - Incluya memoria de estado.
 - Modele lógica completa.
- 20. **Investigación aplicada**
 - Analice un circuito integrado real (ej: 7474, 7490).
 - Evalúe características.
 - Compare con modelo teórico.



4. Aritmética Digital y Sistemas de Procesamiento

4.1 Introducción

La aritmética digital constituye la base del procesamiento de información en sistemas computacionales, microcontroladores y controladores programables. Este capítulo aborda el diseño y análisis de circuitos aritméticos fundamentales, así como su integración en unidades de procesamiento.

4.2 Representación de Números

4.2.1 Sistema binario

$$N = \sum_{i=0}^n b_i 2^i \quad (4.2.1)$$

4.2.2 Complemento a dos

Para un número negativo:

$$N = \bar{N} + 1 \quad (4.2.2)$$

4.2.3 Ejemplo avanzado

Representar -25 en 8 bits:

- $25_{10} = 00011001$
- Inversión: 11100110
- +1: 11100111

4.3 Sumadores

4.3.1 Medio sumador

$$S = A \oplus B \quad (4.3.1)$$

$$C = AB \quad (4.3.2)$$

4.3.2 Sumador completo

$$S = A \oplus B \oplus C_{in} \quad (4.3.3)$$

$$C_{out} = AB + C_{in}(A \oplus B) \quad (4.3.4)$$

4.3.3 Sumador en cascada

$$C_n = f(C_{n-1}) \quad (4.3.5)$$

4.4 Sumador de alta velocidad

4.4.1 Carry Look-Ahead

$$G_i = A_i B_i \quad (4.4.1)$$

$$P_i = A_i \oplus B_i \quad (4.4.2)$$

$$C_{i+1} = G_i + P_i C_i \quad (4.4.3)$$

4.5 Restadores

4.5.1 Restador completo

$$D = A \oplus B \oplus B_{in} \quad (4.5.1)$$

$$B_{out} = \bar{A}B + B_{in}(\bar{A} \oplus B) \quad (4.5.2)$$

4.5.2 Restador mediante complemento a dos

$$A - B = A + (\bar{B} + 1) \quad (4.5.3)$$

4.6 Multiplicadores

4.6.1 Multiplicación binaria

$$P = \sum (A \cdot B_i \cdot i) \quad (4.6.1)$$

4.6.2 Multiplicador combinacional

- Uso de AND y sumadores
- Generación de productos parciales

4.7 División Binaria

4.7.1 Algoritmo básico

- Comparación
- Resta
- Desplazamiento

4.8 Unidad Aritmético-Lógica (ALU)

4.8.1 Estructura

Una ALU realiza operaciones:

- Suma
- Resta
- AND, OR, XOR
- Comparaciones

4.8.2 Modelo matemático

$$F = f(A, B, Control) \quad (4.8.1)$$

4.8.3 Ejemplo avanzado

Diseñar una ALU de 2 bits:

- Entradas: A, B
- Control: selección de operación
- Salidas: resultado y carry

4.9 Análisis de Retardos

$$t_{total} = t_{sumador} + t_{logica} \quad (4.9.1)$$

4.10 Optimización

- Reducción de niveles lógicos
- Uso de lógica paralela
- Minimización de retardos

4.11 Aplicaciones

- Microprocesadores
- DSP
- Controladores industriales
- Sistemas embebidos



Figura 4.12.1: Controladores

4.12 Controladores Programables y Sistemas de Control Digital

4.12.1 Introducción

Los controladores programables constituyen el elemento central en sistemas industriales modernos. Integran procesamiento digital, lógica secuencial y adquisición de señales, permitiendo automatizar procesos complejos.

4.12.2 Arquitectura de un Controlador Programable

Un sistema de control típico incluye:

- Unidad Central de Procesamiento (CPU)
- Memoria (RAM/ROM)
- Módulos de Entrada/Salida (I/O)
- Conversores ADC/DAC

4.12.3 Modelo funcional

$$Salida = f(Entradas, Estado, Programa) \quad (4.12.1)$$

4.12.4 Ciclo de escaneo (Scan Cycle)

El PLC ejecuta continuamente:

1. Lectura de entradas
2. Ejecución del programa
3. Actualización de salidas

4.13 Lógica de Control

4.13.1 Equivalencia con lógica digital

Las funciones implementadas en PLC pueden representarse mediante álgebra booleana:

$$Y = (A \cdot B) + (\bar{C}) \quad (4.13.1)$$

4.13.2 Ejemplo aplicado

Sistema de arranque de motor:

- Botón Start (S)
- Botón Stop (R)
- Sensor de seguridad (E)

Función:

$$Motor = S \cdot \bar{R} \cdot E \quad (4.13.2)$$

4.14 Lenguajes de Programación en PLC

4.14.1 Diagrama Ladder

Representación basada en lógica de relés:

- Contactos (NO/NC)
- Bobinas

4.14.2 Bloques funcionales

Permiten modelar:

- Temporizadores
- Contadores
- Comparadores

4.15 Integración con Sistemas Digitales

4.15.1 Relación con ADC

$$Dato\ digital = ADC(Sensor) \quad (4.15.1)$$

4.15.2 Procesamiento

$$Resultado = ALU(Dato) \quad (4.15.2)$$

4.15.3 Control

$$Salida = f(Resultado) \quad (4.15.3)$$

4.16 Ejemplo Avanzado: Sistema de Control Industrial

Diseñar un sistema que:

- Mida temperatura (ADC)
- Compare con referencia
- Active ventilador
- Active alarma si supera límite crítico

Modelo:

$$T_d = ADC(T) \quad (4.16.1)$$

$$\text{Ventilador} = (T_d > T_{ref}) \quad (4.16.2)$$

$$\text{Alarma} = (T_d > T_{crit}) \quad (4.16.3)$$

4.17 Análisis de Tiempo en Controladores

$$t_{scan} = t_{entrada} + t_{programa} + t_{salida} \quad (4.17.1)$$

4.17.1 Ejemplo

- Entrada: 2 ms
- Programa: 5 ms
- Salida: 3 ms

$$t_{scan} = 10ms \quad (4.17.2)$$

4.18 Modelos Matemáticos de Sistemas de Control

4.18.1 Introducción

El modelado matemático es fundamental para el análisis, diseño y control de sistemas dinámicos. Permite representar el comportamiento de sistemas físicos mediante ecuaciones que describen su evolución en el tiempo.

4.19 Modelado en el Dominio del Tiempo

4.19.1 Ecuaciones diferenciales

Un sistema dinámico puede representarse como:

$$a_n \frac{d^n y(t)}{dt^n} + a_{n-1} \frac{d^{n-1} y(t)}{dt^{n-1}} + \dots + a_0 y(t) = b_m \frac{d^m u(t)}{dt^m} + \dots + b_0 u(t) \quad (4.19.1)$$

4.19.2 Ejemplo: sistema de primer orden

$$\tau \frac{dy(t)}{dt} + y(t) = Ku(t) \quad (4.19.2)$$

4.19.3 Respuesta al escalón

$$y(t) = K \left(1 - e^{-t/\tau} \right) \quad (4.19.3)$$

4.20 Transformada de Laplace

4.20.1 Definición

$$F(s) = \int_0^{\infty} f(t)e^{-st} dt \quad (4.20.1)$$

4.20.2 Función de transferencia

$$G(s) = \frac{Y(s)}{U(s)} \quad (4.20.2)$$

4.20.3 Ejemplo

Para el sistema de primer orden:

$$G(s) = \frac{K}{\tau s + 1} \quad (4.20.3)$$

4.21 Modelos en Espacio de Estados

4.21.1 Forma general

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (4.21.1)$$

$$y(t) = Cx(t) + Du(t) \quad (4.21.2)$$

4.21.2 Ejemplo

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (4.21.3)$$

4.22 Sistemas Discretos

4.22.1 Ecuación en diferencias

$$y[k] = a_1 y[k-1] + b_0 u[k] \quad (4.22.1)$$

4.22.2 Transformada Z

$$G(z) = \frac{Y(z)}{U(z)} \quad (4.22.2)$$

4.23 Estabilidad del Sistema

4.23.1 Criterio

Un sistema es estable si:

- Todos los polos tienen parte real negativa (continuo)
- Todos los polos están dentro del círculo unitario (discreto)

4.24 Controladores

4.24.1 Control proporcional (P)

$$u(t) = K_p e(t) \quad (4.24.1)$$

4.24.2 Control PI

$$u(t) = K_p e(t) + K_i \int e(t) dt \quad (4.24.2)$$

4.24.3 Control PID

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (4.24.3)$$

4.25 Ejemplo Aplicado

Sistema de temperatura:

$$G(s) = \frac{5}{10s + 1} \quad (4.25.1)$$

Control PID:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (4.25.2)$$

4.26 Integración con Sistemas Digitales

4.26.1 Modelo completo

$$\text{Sensor} \rightarrow \text{ADC} \quad (4.26.1)$$

$$\text{ADC} \rightarrow \text{Sistema discreto} \quad (4.26.2)$$

$$\text{Sistema} \rightarrow \text{Controlador} \quad (4.26.3)$$

$$\text{Controlador} \rightarrow \text{Actuador} \quad (4.26.4)$$

4.27 Problemas Propuestos para casa

Los siguientes preguntas propuestas como tareas, cada estudiante, debe entregar realizado a mano en hoja cuadriculada y si realiza las simulaciones ya sea en (Scilab, MATLAB o equivalente). Deberá dibujar a mano y visible.

1. Diseño de sistema de control básico con PLC

- Diseñe un sistema de encendido/apagado de motor usando PLC.
- Modele la lógica en álgebra booleana.
- Dibuje el diagrama Ladder equivalente.

2. Sistema de enclavamiento (Latch)

- Diseñe un sistema Start/Stop con memoria.
 - Modele con flip-flops y Ladder.
 - Analice condiciones de fallo.
3. **Integración ADC + PLC**
 - Diseñe un sistema que lea temperatura analógica.
 - Convierta mediante ADC.
 - Active actuador según umbral.
 4. **Control de nivel de tanque**
 - Diseñe sistema con sensores de nivel.
 - Controle llenado y vaciado.
 - Modele lógica completa.
 5. **Sistema con temporizadores**
 - Diseñe control de motor con retardo de arranque.
 - Implemente temporizador.
 - Analice tiempos.
 6. **Sistema con contadores**
 - Diseñe sistema que cuente piezas en banda.
 - Active alarma al alcanzar límite.
 7. **Análisis de ciclo de escaneo**
 - Modele el scan cycle.
 - Evalúe impacto en sistema rápido.
 8. **Sistema de control de temperatura avanzado**
 - Incluya histéresis.
 - Modele comportamiento.
 - Evite oscilaciones.
 9. **Diseño de sistema de seguridad**
 - Incluya múltiples sensores.
 - Modele lógica de emergencia.
 10. **Integración lógica + PLC**
 - Convierta circuito lógico a Ladder.
 - Compare implementación.
 11. **Diseño de control secuencial**
 - Diseñe secuencia de pasos (ej: llenado, mezcla, descarga).
 - Modele estados.
 12. **Sistema con múltiples condiciones**
 - Diseñe lógica con 4 entradas.
 - Evalúe todas las combinaciones.
 13. **Análisis de fallos en PLC**
 - Identifique fallos posibles.
 - Proponga soluciones.
 14. **Diseño de sistema automático industrial**
 - Modele proceso completo.
 - Integre sensores y actuadores.
 15. **Control con múltiples salidas**
 - Diseñe sistema con 3 actuadores.
 - Coordine funcionamiento.
 16. **Optimización de programa PLC**
 - Reduzca número de instrucciones.
 - Mejore eficiencia.

17. **Sistema con interrupciones**
 - Modele eventos críticos.
 - Analice respuesta.
18. **Simulación completa**
 - Modele sistema en software.
 - Valide comportamiento.
19. **Diseño de sistema embebido equivalente**
 - Compare PLC vs microcontrolador.
 - Diseñe ambos.
20. **Investigación aplicada**
 - Analice un PLC real.
 - Evalúe arquitectura.
 - Compare con modelo teórico.



5. Integración y Proyecto Aplicado

5.1 Introducción

En este capítulo se integran todos los conceptos desarrollados previamente para diseñar un sistema digital completo, desde la adquisición de señales analógicas hasta la toma de decisiones y control de actuadores. Este enfoque es fundamental en aplicaciones industriales, sistemas embebidos y sistemas aeroespaciales.

5.2 Arquitectura General del Sistema

Un sistema digital integrado puede representarse como:

$$\text{Sistema} = \text{Sensor} \rightarrow \text{ADC} \rightarrow \text{Procesamiento} \rightarrow \text{Control} \rightarrow \text{Actuador} \quad (5.2.1)$$

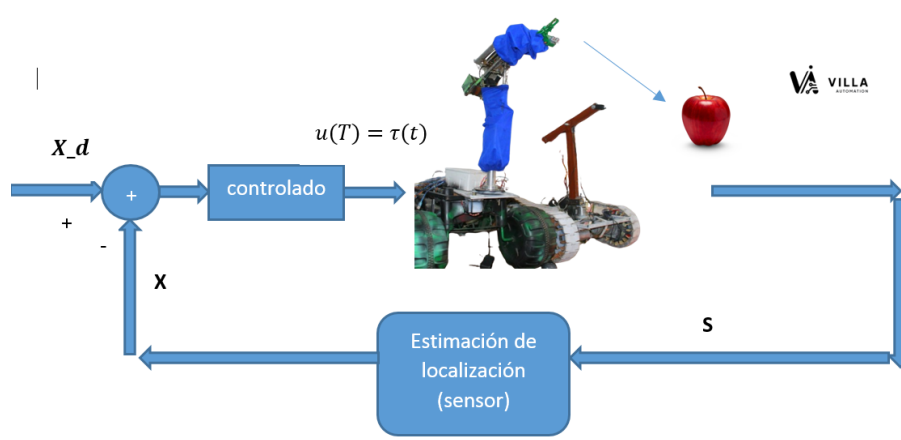


Figura 5.1.1: Integración de proyectos y control

5.2.1 Bloques funcionales

- Sensor (entrada analógica)
- Conversor ADC
- Unidad lógica (combinacional + secuencial)
- Unidad aritmética (ALU)
- Controlador (PLC o microcontrolador)
- Actuador (salida)

5.3 Modelo Matemático del Sistema

$$x(t) \rightarrow ADC \rightarrow x[n] \quad (5.3.1)$$

$$y[n] = f(x[n], Estado[n]) \quad (5.3.2)$$

$$Estado[n+1] = g(x[n], Estado[n]) \quad (5.3.3)$$

$$Salida = h(y[n]) \quad (5.3.4)$$

5.4 Proyecto Integrador: Sistema de Control Inteligente

5.4.1 Descripción

Diseñar un sistema que:

- Mida una variable física (temperatura o presión)
- Digitalice la señal
- Procese la información
- Tome decisiones automáticas
- Active actuadores

5.4.2 Etapas del sistema

Adquisición

$$V_{in} \rightarrow ADC \rightarrow D \quad (5.4.1)$$

Procesamiento

$$Resultado = ALU(D, Referencia) \quad (5.4.2)$$

Control

$$Salida = f(Resultado, Estado) \quad (5.4.3)$$

5.5 Diseño Lógico

5.5.1 Condiciones de control

$$Ventilador = (T > T_{ref}) \quad (5.5.1)$$

$$Alarma = (T > T_{crit}) \quad (5.5.2)$$

5.5.2 Implementación

- Comparadores digitales
- Lógica combinacional
- Flip-flops para estados

5.6 Diseño Secuencial

5.6.1 Máquina de estados

Estados:

- Normal
- Alerta
- Crítico

5.6.2 Transiciones

$$\text{Estado}_{n+1} = f(\text{Estado}_n, \text{Entrada}) \quad (5.6.1)$$

5.7 Implementación en Controlador (PLC)

5.7.1 Lógica Ladder

- Lectura de sensores
- Comparaciones
- Activación de salidas

5.7.2 Ciclo de operación

$$t_{scan} = t_{entrada} + t_{proceso} + t_{salida} \quad (5.7.1)$$

5.8 Simulación del Sistema

5.8.1 Pasos

- Modelar señal analógica
- Simular ADC
- Procesar datos
- Evaluar salida

5.9 Ejemplo Aplicado Avanzado

Diseñar un sistema que:

- Controle temperatura de un sistema crítico
- Active ventilación progresiva
- Active alarma de emergencia

Modelo:

$$T_d = ADC(T) \quad (5.9.1)$$

$$Nivel_1 = (T_d > T_1) \quad (5.9.2)$$

$$Nivel_2 = (T_d > T_2) \quad (5.9.3)$$

$$Alarma = (T_d > T_3) \quad (5.9.4)$$

5.10 Optimización del Sistema

- Reducción de retardos
- Mejora de precisión ADC
- Optimización lógica

5.11 Aplicaciones

- Automatización industrial
- Sistemas embebidos
- Robótica
- Sistemas aeroespaciales

5.12 Aplicación Avanzada: Sistema de Control con Arduino y Robot Colaborativo

5.12.1 Introducción

En esta sección se presenta una aplicación integrada que combina adquisición de datos, procesamiento digital, control automático y actuación en un sistema físico. Se utiliza una plataforma Arduino como controlador embebido y se plantea su extensión hacia un robot colaborativo (cobot).

5.13 Arquitectura del Sistema

El sistema propuesto sigue la estructura:

$$Sensor \rightarrow ADC(Arduino) \rightarrow Procesamiento \rightarrow Control \rightarrow Actuador \quad (5.13.1)$$

5.13.1 Componentes

- Sensor de temperatura (LM35 o similar)
- Arduino (ADC de 10 bits)
- Actuador (ventilador o motor DC)
- Driver (transistor o módulo relé)

5.14 Modelo Matemático

5.14.1 Conversión ADC

$$T = \frac{V_{adc} \cdot 5}{1024 \cdot 0,01} \quad (5.14.1)$$

5.14.2 Control ON/OFF con histéresis

$$u = 1 \quad \text{si } T > T_{max} \quad (5.14.2)$$

$$u = 0 \quad \text{si } T < T_{min} \quad (5.14.3)$$

5.15 Implementación en Arduino

5.15.1 Código básico

```

const int sensorPin = A0;
const int motorPin = 9;

float T;
float T_max = 30.0;
float T_min = 25.0;

bool estado = false;

void setup() {
  pinMode(motorPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int adc = analogRead(sensorPin);
  float voltage = adc * (5.0 / 1024.0);
  T = voltage / 0.01;

  if (T > T_max) {
    estado = true;
  }
  if (T < T_min) {
    estado = false;
  }

  digitalWrite(motorPin, estado);

  Serial.print("Temperatura: ");
  Serial.println(T);

  delay(500);
}

```

5.16 Extensión: Control PID en Arduino

5.16.1 Modelo discreto

$$u[k] = K_p e[k] + K_i \sum e[k] + K_d (e[k] - e[k - 1]) \quad (5.16.1)$$

5.16.2 Código simplificado

```
float Kp = 2.0, Ki = 0.5, Kd = 1.0;
float e, e_prev = 0, integral = 0;

void loop() {
  int adc = analogRead(sensorPin);
  float voltage = adc * (5.0 / 1024.0);
  float T = voltage / 0.01;

  float ref = 28.0;
  e = ref - T;

  integral += e;
  float deriv = e - e_prev;

  float u = Kp*e + Ki*integral + Kd*deriv;

  analogWrite(motorPin, constrain(u, 0, 255));

  e_prev = e;
  delay(100);
}
```

5.17 Aplicación en Robot Colaborativo

5.17.1 Descripción

El sistema puede escalarse a un robot colaborativo donde:

- Sensores monitorean variables (temperatura, fuerza, posición)
- Arduino o controlador procesa señales
- Actuadores ejecutan acciones seguras

5.17.2 Modelo del sistema

$$x[k+1] = Ax[k] + Bu[k] \quad (5.17.1)$$

$$y[k] = Cx[k] \quad (5.17.2)$$

5.18 Ejemplo: Control de Posición

$$\theta_{ref} - \theta = e(t) \quad (5.18.1)$$

$$u(t) = PID(e(t)) \quad (5.18.2)$$

5.19 Código simplificado (Servo)

```
#include <Servo.h>

Servo motor;

int sensorPin = A0;

void setup() {
  motor.attach(9);
}

void loop() {
  int val = analogRead(sensorPin);
  int angle = map(val, 0, 1023, 0, 180);
  motor.write(angle);
  delay(20);
}
```

5.20 Seguridad en Robots Colaborativos

- Límites de fuerza
- Sensores redundantes
- Paradas de emergencia

5.21 Conclusión

La integración de Arduino con sistemas de control permite implementar soluciones reales de bajo costo. La extensión hacia robots colaborativos demuestra la aplicación directa de los conceptos desarrollados en este libro en sistemas industriales modernos.

5.22 Problemas Propuestos para casa

Los siguientes preguntas propuestas como tareas, cada estudiante, debe entregar realizado un artículo (usar estilo JICAAT <https://www.villautomation.net/>) y un proyecto basado en lo aprendido.

1. **Sistema completo de adquisición y control**
 - Diseñe un sistema: Sensor → ADC → Procesamiento → Control → Actuador.
 - Modele matemáticamente cada etapa.
 - Simule el sistema completo.
2. **Control inteligente de temperatura industrial**
 - Diseñe control con múltiples niveles.
 - Incluya histéresis.
 - Implemente lógica secuencial.
3. **Sistema de monitoreo multivariable**
 - Integre 3 sensores (temperatura, presión, nivel).
 - Diseñe lógica de decisión.
 - Active diferentes actuadores.
4. **Diseño de sistema embebido completo**
 - Implemente en microcontrolador.
 - Integre ADC y lógica digital.
 - Compare con PLC.

5. **Sistema de control de motor inteligente**
 - Diseñe arranque progresivo.
 - Incluya protección.
 - Modele comportamiento dinámico.
6. **Implementación de control PID en Arduino**
 - Diseñe controlador PID.
 - Ajuste parámetros.
 - Analice estabilidad.
7. **Sistema de monitoreo multivariable**
 - Integre 3 sensores analógicos.
 - Procese señales en Arduino.
 - Diseñe lógica de control.
8. **Control de motor con PWM**
 - Diseñe control de velocidad.
 - Modele sistema dinámico.
 - Implemente en Arduino.
9. **Sistema de control con histéresis optimizada**
 - Diseñe control estable.
 - Evite oscilaciones.
 - Analice comportamiento.
10. **Sistema de adquisición de alta precisión**
 - Mejore resolución ADC.
 - Analice errores.
 - Proponga mejoras.
11. **Diseño de robot colaborativo básico**
 - Controle un servo.
 - Use sensor analógico.
 - Modele sistema.
12. **Control de posición con PID (servo)**
 - Modele sistema.
 - Implemente control.
 - Evalúe error.
13. **Sistema de seguridad en robot colaborativo**
 - Diseñe detección de obstáculo.
 - Implemente parada automática.
14. **Control secuencial con Arduino**
 - Diseñe máquina de estados.
 - Controle múltiples actuadores.
15. **Integración ADC + control + actuador**
 - Modele flujo completo.
 - Implemente sistema funcional.
16. **Integración PLC + lógica digital**
 - Diseñe sistema híbrido.
 - Compare implementación.
17. **Sistema de adquisición de alta precisión**
 - Diseñe ADC de alta resolución.
 - Analice errores.
18. **Diseño de sistema distribuido**
 - Modele múltiples módulos.

- Integre comunicación.
19. **Sistema de control en tiempo real**
 - Analice restricciones de tiempo.
 - Diseñe solución.
 20. **Aplicación aeroespacial**
 21. **Integración completa del sistema de vuelo**
 - Integre sensores, control y actuadores.
 - Modele sistema completo.
 22. **Simulación de sistema aeroespacial**
 - Modele sistema en software.
 - Compare resultados.
 23. **Optimización de sistema embarcado**
 - Reduzca consumo.
 - Mejore eficiencia.
 24. **Sistema de navegación básico**
 - Use sensores.
 - Estime posición.
 - Modele sistema.
 25. **Diseño de computadora de vuelo (simplificada)**
 - Integre todos los módulos.
 - Modele funcionamiento.



Bibliografía

Books

5.23 Bibliografía Recomendada

- TOCCI, R.J., WIDMER, N.S. y MOSS, G.L. (2011) *Sistemas Digitales: Principios y Aplicaciones*. Pearson. Capítulos 1–6
- FLOYD, T.L. (2011) *Fundamentos de Sistemas Digitales*. Pearson. Capítulos 2–7
- MANO, M.M. y CILETTI, M.D. (2013) *Digital Design*. 5ª edición. Pearson. Capítulos 1–5
- WAKERLY, J.F. (2005) *Digital Design: Principles and Practices*. 4ª edición. Pearson. Capítulos 2–6
- OGATA, K. (2010) *Modern Control Engineering*. 5ª edición. Prentice Hall. Capítulos 2–6
- NISE, N.S. (2011) *Control Systems Engineering*. 6ª edición. Wiley. Capítulos 3–7
- FRANKLIN, G.F., POWELL, J.D. y EMAMI-NAEINI, A. (2015) *Feedback Control of Dynamic Systems*. 7ª edición. Pearson. Capítulos 4–8
- ASTROM, K.J. y WITTENMARK, B. (1997) *Computer-Controlled Systems*. 3ª edición. Prentice Hall. Capítulos 3–6

- BOLTON, W. (2015) *Programmable Logic Controllers*. 6ª edición. Elsevier. Capítulos 1–5
- PETRUZELLA, F.D. (2016) *Programmable Logic Controllers*. 5ª edición. McGraw-Hill. Capítulos 2–6
- BANZI, M. y SHILOH, M. (2014) *Getting Started with Arduino*. 3ª edición. Maker Media. Capítulos 1–4
- MONK, S. (2017) *Programming Arduino: Getting Started with Sketches*. McGraw-Hill. Capítulos 2–6
- SICILIANO, B., SCIAVICCO, L., VILLANI, L. y ORIOLO, G. (2009) *Robotics: Modelling, Planning and Control*. Springer. Capítulos 2–7
- CRAIG, J.J. (2004) *Introduction to Robotics: Mechanics and Control*. 3ª edición. Pearson. Capítulos 3–6
- SUTTON, G.P. y BIBLARZ, O. (2010) *Rocket Propulsion Elements*. 8ª edición. Wiley. Capítulos 1–5
- STEVENS, B.L., LEWIS, F.L. y JOHNSON, E.N. (2015) *Aircraft Control and Simulation*. 3ª edición. Wiley. Capítulos 2–6
- FORTESCUE, P., STARK, J. y SWINERD, G. (2011) *Spacecraft Systems Engineering*. 4ª edición. Wiley. Capítulos 3–7
- GREWAL, M.S. y ANDREWS, A.P. (2015) *Kalman Filtering: Theory and Practice*. 4ª edición. Wiley. Capítulos 1–5
- SIMON, D. (2006) *Optimal State Estimation*. Wiley. Capítulos 1–4
- HUAYANAY VILLAR, J.L., BEIZAGA REYES, F.X., SOTELO GUTIÉRREZ, C., AYALA CONDORI, K.E. y MENESES HIYO, S.Y. (2025) *Resultados experimentales del cohete aerospacial JICAAT, Ayacucho-Perú*. Revista de Investigación Científica de la UNF – Aypate, 4(2), 85–103.

DOI: 10.57063/ricay.v4i2.164

- HUAYANAY VILLAR, J.L., CORREA CHILON, D. y ZAMBRANO CARRERA, D.H. (2024) *Implementación de un programa nacional de actividades espacial del proyecto JI-CAAT, Perú, 2023–2032*. Revista de Investigación Científica de la UNF – Aypate, 3(1), 79–91.

DOI: 10.57063/ricay.v3i1.81

- HUAYANAY VILLAR, J.L. (2024) *Seguimiento de trayectoria en robótica móvil con alimentación de energía solar*. Revista de Investigación Científica de la UNF – Aypate, 2(4), 1–12.

DOI: 10.57063/ricay.v2i4.64

- HUAYANAY VILLAR, J.L. y MENESES HIYO, S.Y. (2024) *Diseño y construcción de un sistema inteligente de control de bombas de agua en cultivo de fresas (Fragaria)*. Revista de Investigación Científica de la UNF – Aypate, 2(4), 24–35.

DOI: 10.57063/ricay.v2i4.66

- HUAYANAY VILLAR, J.L. y MENESES HIYO, S.Y. (2025) *Diseño y Control Predictivo de un Robot Colaborativo para Aplicaciones en Agricultura y Minería*. Revista de Investigación Científica de la UNF – Aypate, 4(1), 37–49.

Acceso al artículo

- HUAYANAY VILLAR, J.L. et al. (2025) *Cohete Automatizado Reutilizable JICAAT*. Libro de Resúmenes ECI 2025.

Acceso al documento

- HUAYANAY VILLAR, J.L. (2026) *Robots colaborativos aplicados a sistemas de defensa*. Libro de Resúmenes ECI 2026.

Acceso al documento

- HUAYANAY VILLAR, J.L. y CUNHA, J.P.V.S. (2022) *Sliding-Mode Controllers for Soil Moisture Modeled by Nonlinear Parabolic Partial Differential Equations*. IEEE Variable Structure Systems (VSS).

DOI: 10.1109/VSS57184.2022.9902067

