UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN Y DISEÑO DIGITAL

TEMA

ARITMÉTICA DE LA COMPUTADORA Y ÁLGEBRA DE BOOLE.

INTEGRANTES

ALVIA VILLEGAS ERICK ADALBERTO – Hizo un mal trabajo, parte de su trabajo lo que tenía en su texto no concordaba con las fuentes – No realizó su parte del apartado de procedimientos.

FALCONES RODRIGUEZ LUIS DAVID - Hizo un mal trabajo, parte de su trabajo lo que tenía en su texto no concordaba con las fuentes – No realizó su parte del apartado de procedimientos.

MORALES SANCHEZ GARY ALEJANDRO

NIEVES SANCHEZ JIMMY SAMUEL

PONCE RIVERA MERY HELENMEY

CURSO

2DO SOFTWARE “B”

GRUPO

I

MATERIA

ARQUITECTURA DE COMPUTADORAS

INDICE

1. OBJETIVO

Aplicar conceptos de aritmética binaria y decimal, álgebra booleana y circuitos lógicos para comprender el funcionamiento de los microprocesadores, microcontroladores y sistemas digitales.

2. FUNDAMENO TEÓRICO

2.1 Aritmética Binaria y Decimal

Parte de FALCONES RODRIGUEZ LUIS DAVID

2.2 Álgebra de Circuitos Digitales y Función Boole

Parte de ALVIA VILLEGAS ERICK ADALBERTO

2.3 Circuitos Lógicos y Sistemas Digitales

Zhang (2025) menciona que los circuitos lógicos y sistemas digitales son partes fundamentales en el diseño de sistemas embebidos y en algunos aparatos electrónicos modernos[1]. Se utilizan también para compartir información en forma de señales digitales como 0 y 1 [2]. Donzellini et al. (2018) indican que esas señales son fundamentales para el uso de las compuertas lógicas como AND, OR, NOT, XOR, NAND y NOR. [3].

2.3.1 Circuitos Combinacionales

Lian et al. (2024) señalan que en los circuitos combinacionales la salida depende del valor actual de la entrada en un momento específico. Esos circuitos no contienen ningún tipo de elemento que almacene[4] Ashok y Tripura Sundari (2023) explican que, es decir, solamente se activa la salida cuando recibe valor al instante y no de valores anteriores. [5].

Sumadores

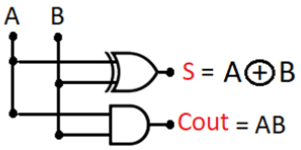
Amini Valashani et al. (2018) afirman que un sumador es un circuito lógico que se utiliza en la operación básica que es la suma, y es fundamental para la parte de las Unidades Aritmético-Lógicas (ALU) [6] Pritty (2024) indica que existen 2 tipos de sumadores: el Sumador Half Adder y el Sumador Full Adder. [7].

Sumador Half Adder (Sumador de medio)

KasraAzizbeigi (2021) explican que estos consisten en que se ingresan 2 bits; al sumar los 2 bits se obtienen 2 salidas: la primera es Suma y la segunda es Acarreo. Tiene una salida para mostrar el resultado y otra que define el acarreo en caso de existir [8].

Sumador

A	B	S	Cout
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

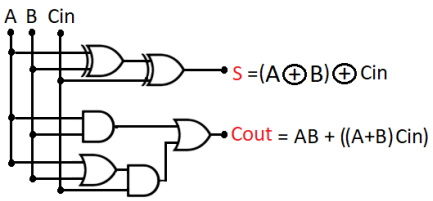


Sumador Full Adder (Sumador Completo)

Vahabi et al. (2021) también indican que se realiza la suma de tres bits: A, B y el acarreo de entrada (Cin). Entrega una suma (S) y un acarreo de salida (Cout) [9].

Sumador completo

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Restadores

Ye et al. (2023) mencionan que los restadores son circuitos digitales que obtienen similitud con los sumadores, pero en vez de sumar se restan las entradas de datos binarios[10]. **Ghadi (2021)** señala que, igual que los sumadores, existen 2 tipos como el Half Subtractor y Full Subtractor. [11].

Restador de Medio (Half Subtractor)

Mirizadeh y Asghari (2022) explican que el restador de medio es también conocido como un circuito combinacional que se realiza con 2 entradas de datos de dos bits

binarios: un minuendo y un sustraendo[12]. **Surendra y Reddy (2022)** indican que existen 2 tipos de salida: la primera se llama Diferencia y la segunda Préstamo (Borrow), eso ocurre cuando se necesita pedir prestado de un bit de orden superior [13].

A	B	Diferencia	Préstamo
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Multiplexor

Parandin et al. (2024) afirman que el circuito lógico multiplexor tiene la funcionalidad de seleccionar una de varias entradas de selección de datos y tener una sola salida [14]. **Parandin y Bagheri (2023)** explican que también sirven para la funcionalidad de un dispositivo de conmutación, procesando señales de comunicación [15].

Características clave de un multiplexor:

Entrada

Ghafouri y Manavizadeh (2021) indican que un circuito multiplexor tiene múltiples entradas y se va etiquetando cada una de las entradas con la letra I_0 hasta la última entrada [16].

Señales de control

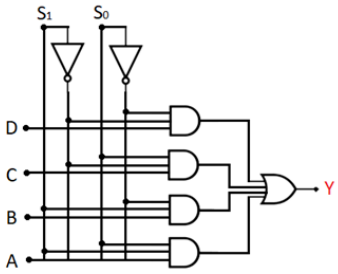
Sooriamala et al. (2023) señalan que esas señales de control se utilizan para determinar la conexión de entrada a la salida y se denominan como S_1 y S_0 [17].

Kumar et al. (2025) concluyen que solo una de las entradas se selecciona para ser enviada a la salida, según las señales de control [18].

Multiplexor 4 a 1

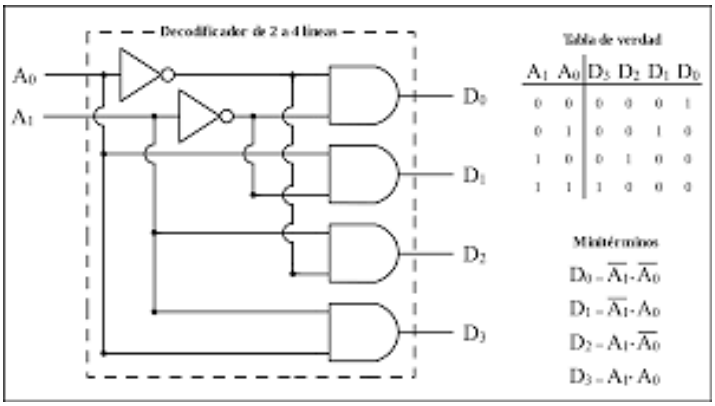
S1	S0	A	B	C	D	Y
0	0	x	x	x	0	0
0	0	x	x	x	1	1
0	1	x	x	0	x	0
0	1	x	x	1	x	1
1	0	x	0	x	x	0
1	0	x	1	x	x	1
1	1	0	x	x	x	0
1	1	1	x	x	x	1

$Y = S_1S_0A + S_1\overline{S_0}B + \overline{S_1}S_0C + \overline{S_1}\overline{S_0}D$



Decodificador

Xia et al. (2024) mencionan que el circuito lógico decodificador tiene la función de recibir un código binario codificado y se activa exactamente una de sus múltiples salidas [19] **Zahoor et al. (2024)** explican que la salida única se activa cuando al decodificar esos valores son correspondientes a la entrada [20].



Comparador

Majeed et al. (2021) indican que el circuito lógico comparador combinacional tiene como función comparar los números binarios para ver si son iguales, mayores o menores, con la finalidad de determinar su relación, tomar decisiones y ordenar datos [21]. **Darbandi et al. (2024)** explican que dentro del circuito comparador existen 3 tipos: el primero es Comparador de Igualdad (Equal Comparator), el segundo es Comparador de Magnitud (Magnitude Comparator) y el tercero Comparador Completo (Full Comparator) [22].

Comparador de igualdad (Equal comparator)

Donaire et al. (2024) dicen que el comparador de igualdad sirve para distinguir si los valores de entrada binarios son iguales. Si ambos valores son semejantes, el valor de la salida será alto; mientras que si no son equivalentes, será bajo [23].

Comparador de magnitud (Magnitudes Comparator)

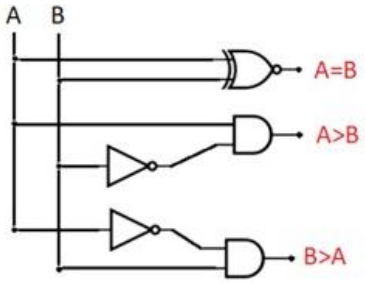
Saha et al. (2021) aclaran que el comparador de magnitud ayuda a distinguir de los 2 valores binarios cuál es el mayor o menor [24].

Comparador completo (Full Comparator)

Jendernalik (2017) añade que el comparador completo o full comparator obtiene ambas funcionalidades del comparador de igualdad y del de magnitud [25].

Comparador

A	B	A=B	A>B	B>A
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0



2.3.2 Circuitos Lógicos Secuenciales

Singh Kalyan et al. (2024) explican que los circuitos lógicos secuenciales son lo inverso a los circuitos combinacionales, ya que no solamente dependen de las entradas actuales, sino que toman en cuenta los valores anteriores [26]. **Niu et al. (2025)** indican que la salida del circuito secuencial, para que se active, necesita recordar el estado anterior. Esta memoria se puede implementar utilizando elementos como flip-flops o registros [27].

flip-flops

Alharbi et al. (2023) explican que en esos circuitos lógicos secuenciales se puede almacenar información de solamente 1 bit, claramente que es una memoria temporal [28]. En estos circuitos solamente puede estar en 2 estados, 0 o 1, y cambia su estado en respuesta a una señal de control llamada reloj[29].

Tipos de flip-flops

Set-Reset

Vali y Kumar (2024) mencionan que el tipo **Set-Reset** tiene la funcionalidad de almacenar memoria, pero es muy limitada, ya que solamente guarda un bit [30] En este circuito tiene 2 entradas: set, que es poner 1, y reset, que es poner 0 [31].

Data

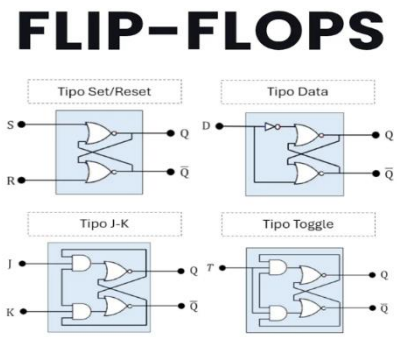
Bashir et al. (2023) explican que en el tipo **Data**, el circuito recibe 1 valor de entrada y tiene 2 salidas: una es alta y la otra baja [32].

JK

Lin et al. (2022) indican que el **JK** ayuda a eliminar los datos inválidos y tiene la capacidad de almacenar y poner 1 o 0, muy parecido al set-reset [33].

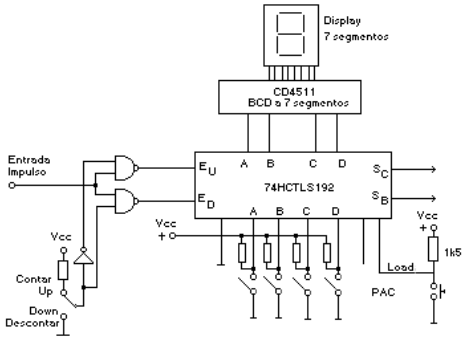
Toggle

Alharbi et al. (2023) explican que, en el **Toggle**, el valor que ingresa al salir cambia su resultado, o sea, se invierte con cada pulso del reloj [34].



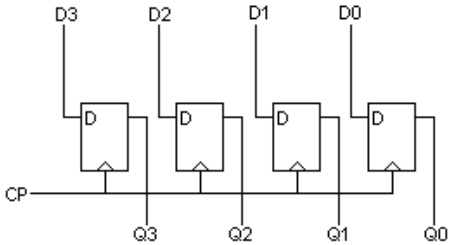
Contador

Abutaleb (2017) dice que en ese circuito se ayuda a registrar el número de pulsos o los eventos que recibe en su entrada del reloj. Si no recibe valor, se mantiene, y si recibe nuevo valor, se cambia [35].



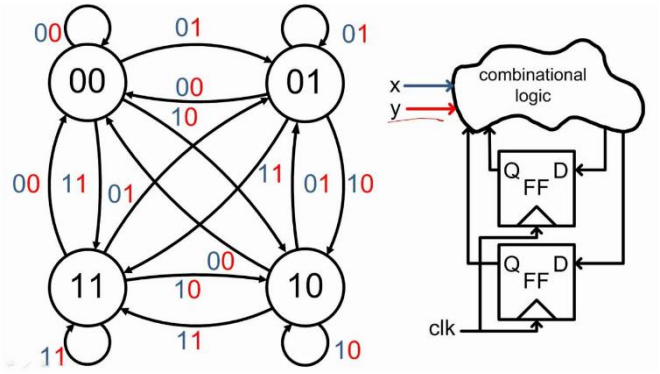
Registro

Gholamnia Roshan y Gholami (2024) indican que el registro es un circuito que está compuesto por un conjunto de bits y ayuda a almacenar más información y transferir datos binarios [36].



Máquina de estados

Mohmed et al. (2020) afirman que una máquina de estados tiene una función importante, ya que puede encontrarse en varios estados posibles y cambia de estado en función de las entradas y el reloj[37]. Sirve para que respondan a los eventos de entrada, tome decisiones condicionales y ayude al control de procesos en función del estado actual [37].



2.4 Microprocesador y Microcontroladores

2.4.1 Características básicas de los microprocesadores y microcontroladores

Para Darche Philippe (2020), en casi todas las partes del mundo se encuentran sistemas digitales con al menos un microprocesador o microcontrolador. Estos componentes se encuentran en lavadoras, autos, cámaras, equipos médicos, consolas de videojuegos, celulares, servidores, entre otros más [38].

Dicho por Rankovska Valentina (2021), las placas de microcontroladores de código abierto no suelen incorporar periféricos de usuario. Estas placas pueden complementarse con módulos que permitan ampliar sus funciones, como sensores, motores eléctricos, módulos de entrada y salida, entre otros [39].

En el libro "Computation Model and Architecture: Illustration with the von Neumann Approach" (2020), se explica que las diferencias entre microprocesadores y microcontroladores se basan en la estructura y organización. Los microprocesadores requieren memoria RAM externa, mientras que los microcontroladores incluyen memoria RAM interna dentro del mismo chip [38].

Los microprocesadores suelen operar a frecuencias de varios GHz (por ejemplo, el AMD Ryzen 7 3700X [40], el Intel Core i5-12600k [41] y el Intel Pentium 4405U [42]), mientras que los microcontroladores trabajan mayoritariamente en el rango de MHz (como el PIC16F87XA [43], el ATmega328P [44] o el MSP430FR6005 [45])

2.4.2 Velocidad de procesamiento y su ciclo de instrucciones de microprocesadores.

2.4.2.1 AMD Ryzen 7 3700X

Según el sitio web oficial de AMD, el microprocesador Ryzen 7 3700X tiene una frecuencia base de 3.6GHz y una frecuencia máxima de 4.4GHz. Debido a sus diferentes frecuencias, sea la base o la máxima, sus duraciones de un ciclo de instrucción está entre 0.227ns y 0.294ns [40].

2.4.2.2 Intel Core i5-12600k

Como se indica en el sitio web oficial de Intel, el microprocesador Intel Core i5-12600k dispone de una frecuencia base de 2.80GHz y una frecuencia máxima de 4.90GHz, esto quiere decir que, tiene duraciones de un ciclo de instrucción entre 0.357ns y 0.204ns [41].

2.4.2.3 Intel Pentium 4405U

En el sitio web oficial de Intel, el microprocesador Intel Pentium 4405U tiene una frecuencia de 2.10GHz, por lo que su duración de un ciclo de instrucción es de 0.476ns [42]

2.4.3 Velocidad de procesamiento y su ciclo de instrucciones de microcontroladores

2.4.3.1 PIC16F87XA

Según el sitio web oficial de “Microchip”, este microcontrolador dispone de una frecuencia máxima de 20MHz y 200ns de duración de un ciclo de instrucción [43].

2.4.3.2 ATmega328P

Como se indica en el sitio web oficial de “Microchip”, El ATmega328P tiene una frecuencia máxima de 16MHz, puede alcanzar hasta los 16MIPS (millones de instrucciones por segundo), y tiene 62.5ns de duración de un ciclo de instrucción [44].

2.4.3.3 MSP430FR6005

En la documentación oficial del microcontrolador MSP430FR6005 indica que este posee una frecuencia máxima de 16MHz, y tiene un tiempo de ejecución de registro a registro de un ciclo de reloj, es decir que su duración por ciclo de instrucción es de 62.5ns [45].

3. PROCEDIMIENTOS

3.1 Operaciones de Aritmética Binaria y Decimal

Parte de FALCONES RODRIGUEZ LUIS DAVID

3.1.1 Realizar ejercicios de suma, resta, multiplicación y división en binario y decimal

Parte de FALCONES RODRIGUEZ LUIS DAVID

3.1.2 Comparar resultados obtenidos en ambos sistemas para evaluar la precisión

Parte de FALCONES RODRIGUEZ LUIS DAVID

3.2 Ejercicios de Álgebra Booleana

Parte de ALVIA VILLEGAS ERICK ADALBERTO

3.2.1 Simplificar expresiones usando mapas de Karnaugh

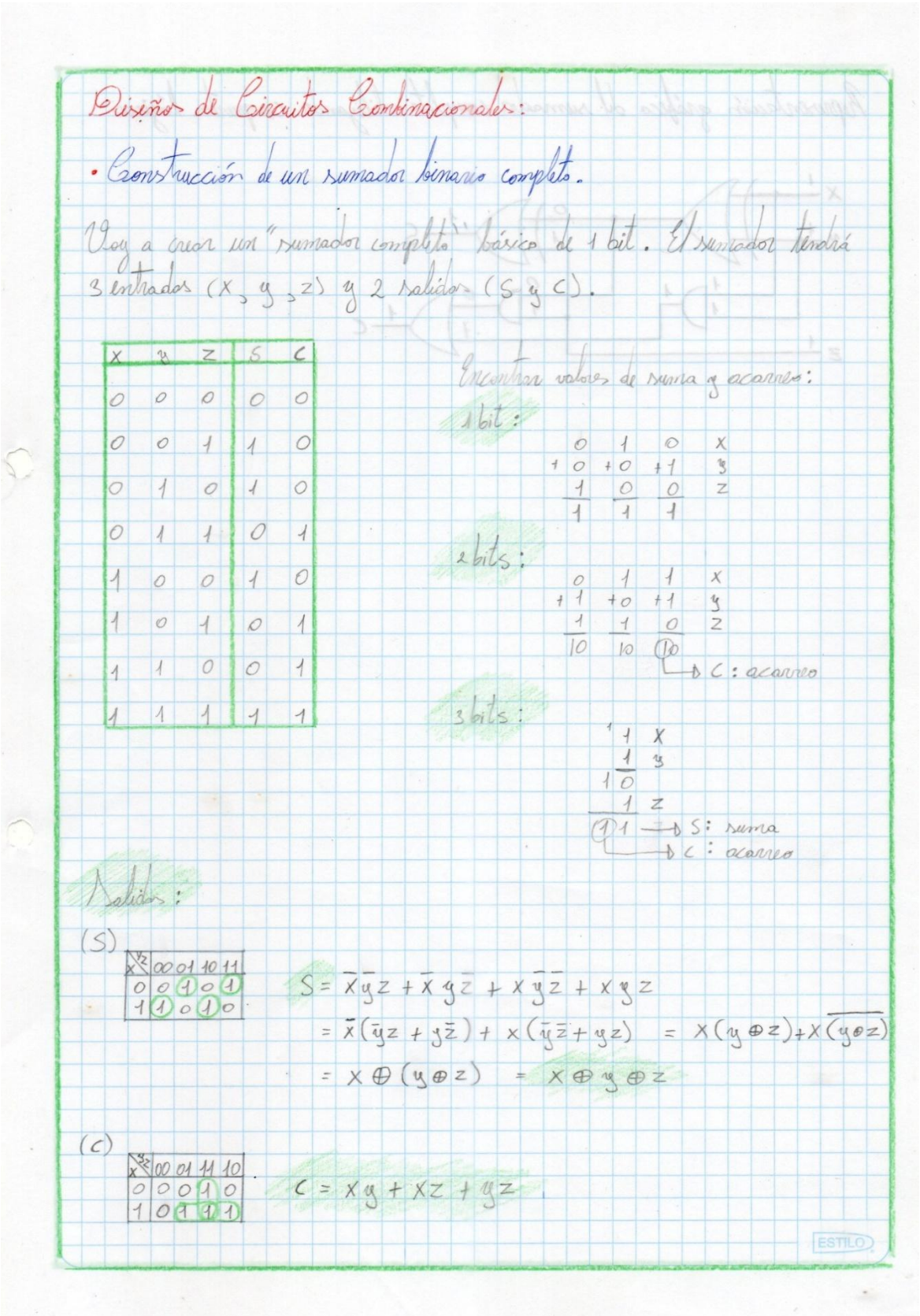
Parte de ALVIA VILLEGAS ERICK ADALBERTO

3.2.2 Diseñar y probar circuitos lógicos básicos utilizando software de simulación

Parte de ALVIA VILLEGAS ERICK ADALBERTO

3.3 Diseño de Circuitos Combinacionales

3.3.1 Construcción de un sumador binario completo



Diseños de Circuitos Combinacionales:

• Construcción de un sumador binario completo.

Voy a crear un "sumador completo" básico de 1 bit. El sumador tendrá 3 entradas (x, y, z) y 2 salidas (S y C).

x	y	z	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Encontrar valores de suma y acarreo:

1 bit:

$$\begin{array}{r} 0\ 1\ 0\ \ x \\ +\ 0\ +0\ +1\ \ y \\ \hline 1\ \ 0\ \ 0\ \ z \\ 1\ \ 1\ \ 1 \end{array}$$

2 bits:

$$\begin{array}{r} 0\ 1\ 1\ \ x \\ +\ 1\ +0\ +1\ \ y \\ \hline 1\ \ 1\ \ 0\ \ z \\ 10\ \ 10\ \ (10) \end{array}$$

→ C: acarreo

3 bits:

$$\begin{array}{r} {\scriptstyle 1}\ 1\ x \\ \ \ 1\ y \\ \hline 1\ 0 \\ \ \ 1\ z \\ \hline (1)1 \end{array}$$

→ S: suma

→ C: acarreo

Salidas:

(S)

x \ yz	00	01	10	11
0	0	1	0	1
1	1	0	1	0

$$S = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$$

$$= \bar{x}(\bar{y}z + y\bar{z}) + x(\bar{y}\bar{z} + yz) = x(y \oplus z) + x\overline{(y \oplus z)}$$

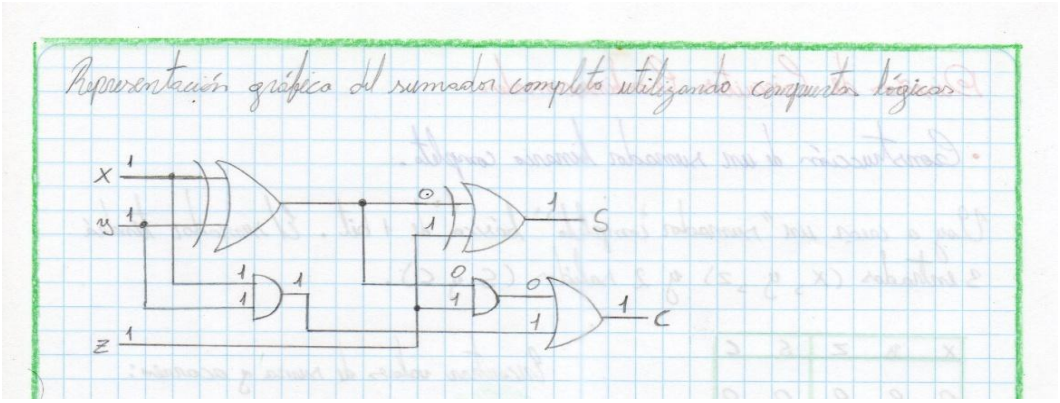
$$= x \oplus (y \oplus z) = x \oplus y \oplus z$$

(C)

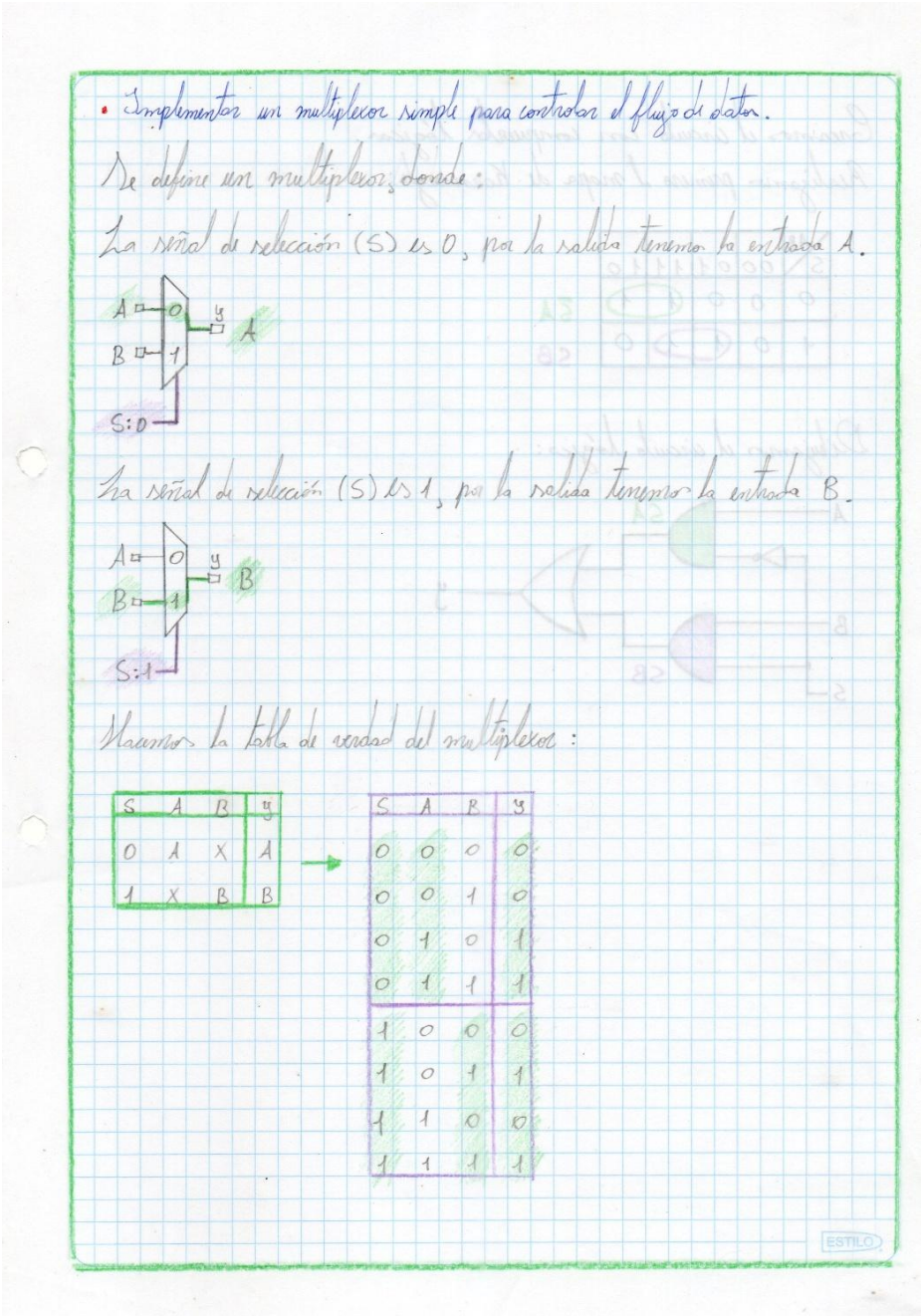
x \ yz	00	01	11	10
0	0	0	1	0
1	0	1	1	1

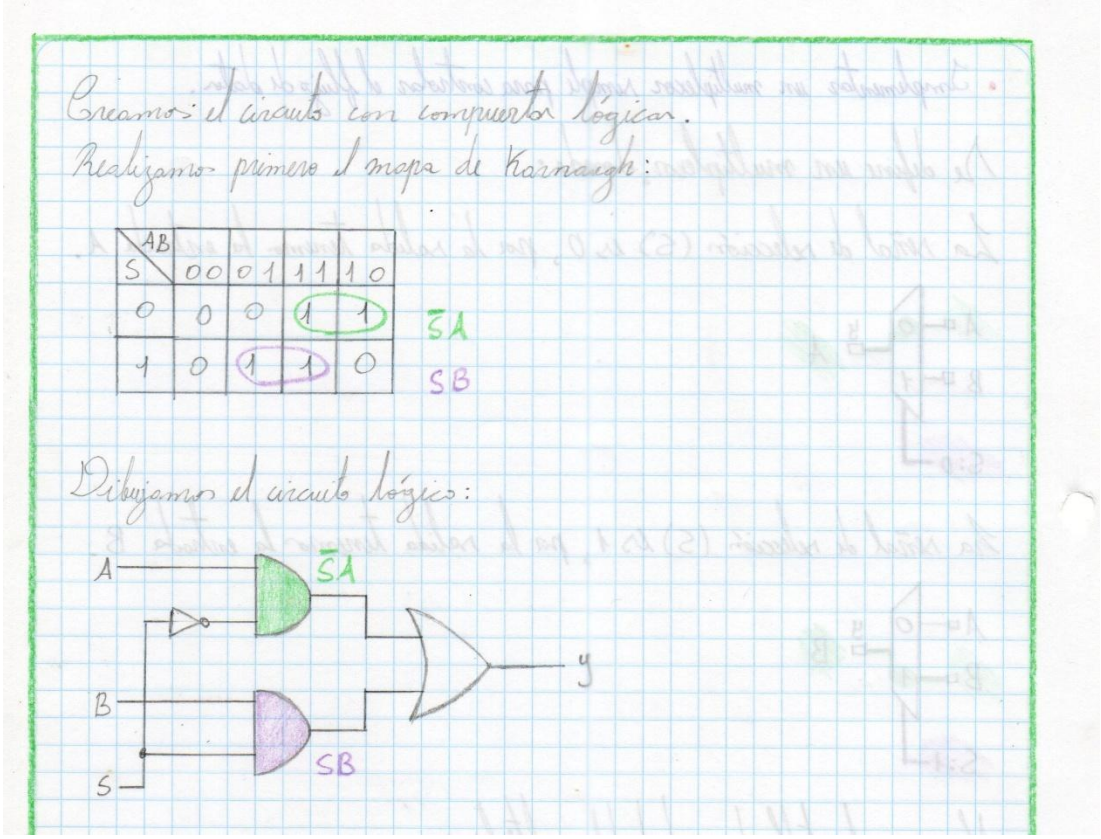
$$C = xy + xz + yz$$

ESTILO



3.3.2 Implementar un multiplexor simple para controlar el flujo de datos

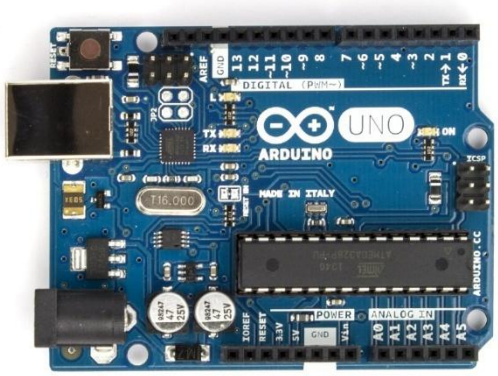




3.4 Estudio de Microprocesadores y Microcontroladores

3.4.1 Investigar la arquitectura básica de un microprocesador y un microcontrolador

Según “Computation Model and Architecture: Illustration with the von Neumann Approach” (2025), el desarrollo de los microprocesadores y microcontroladores ha sido determinante en la evolución de la computación moderna, al ofrecer soluciones integradas que permiten procesar información de manera eficiente y en tiempo real. Estos componentes son fundamentales para la arquitectura, tiene el propósito de dispositivos inteligentes. Lo microprocesadores están diseñados para ejecutar operaciones complejas, involucrando ciclos de instrucción optimizados y sistemas de gestión de memoria avanzados, como las caches multinivel [38].



Según Philippe Darche (2025), por su parte, los microcontroladores se presentan como soluciones altamente integradas y económicas para tareas específicas. Estos incluyen CPU, RAM, ROM, puertos de entrada/salida, convertidores y temporizadores, todo contenido en un solo chip; su uso es común en automatización industrial, sistemas embebidos, control de sensores y aplicaciones de Internet de las cosas, dado su bajo consumo energético y su capacidad de operar de manera automática [38].

3.4.1.1 Arquitectura básica de un microprocesador

Según Hübner y Becker (2011), la arquitectura de un microprocesador moderno consiste en llevar a cabo múltiples núcleos, cada uno con su propia caché L1 y L2 locales, mientras comparten niveles superiores de caché mediante protocolos de coherencia distribuidos. Esto permite que los datos se muevan eficientemente entre núcleos y memoria, esto ayuda a ir mejorando el rendimiento y la escalabilidad en sistemas multicore. Un ejemplo completo de este tipo de diseño se presenta en la literatura sobre Multiprocessor System-On-Chip, donde se describe la organización de cache y la coherencia en arquitecturas contemporáneas[46].

Según Calimera et al (2021), además, algunos procesadores modernos utilizan arquitecturas con redes internas o Network.on.Chip (NOC) para poder conectar los núcleos y subsistemas de memoria/periféricos. Las NoC permiten múltiples flujos de

datos simultáneos de alta velocidad, evitando los cuellos de botella de los buses tradicionales y mejorando la comunicación interna del sistema[47].

CICLO DE INTRUCCIONES Y VELOCIDAD

Según Tang et al (2025), el de instrucción feetch, decode, execute, memory access, write-back; se potencia mediante técnicas de pipelining que permiten solapar las etapas de múltiples instrucciones. Algunas herramientas modernas simulan el comportamiento de pipeline a nivel ciclo, mejorando la precisión de análisis de rendimiento mediante frameworks como PipeDBT, que modelan formalmente los estados de pipeline y permiten simular arquitecturas VLIW con precisión de ciclos [48]

Según Rober J. Colvin (2025) dice que dado que la ejecución fuera de orden (out-of-order) y la especulación pueden dejar residuos en cache que representan canales de fuga de información, es esencial modelar formalmente estos comportamientos. Se han desarrollado modelos semánticos estructurales que permiten comprender estas vulnerabilidades y diseñar procesadores seguros y eficientes simultáneamente [49].

- **Búsqueda (Fetch):** El microprocesador recupera la instrucción desde la memoria principal (RAM), accediendo a la dirección indicada por el contador de programa (PC).
- **Decodificación (Decode):** La unidad de control interpreta la instrucción para identificar la operación que debe ejecutarse y los operandos involucrados.
- **Ejecución (Execute):** Se realiza la operación especificada (suma, carga, salto, etc.).
- **Escritura del resultado (Write-back):** El resultado se guarda en un registro o en memoria.

Formula general para el tiempo de ejecución de un programa:

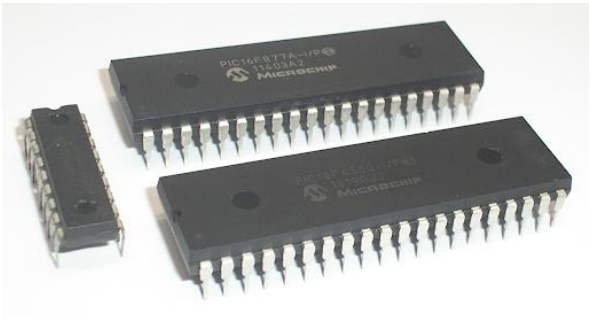
$$Tiempo\ de\ ejecucion = \frac{N^{o}\ de\ intrucciones\ \times Ciclos\ por\ instruccion}{Frecuencia\ del\ reloj}$$

3.4.1.2 Arquitectura básica de un microcontrolador

Según Proceedings of the ACM/IEEE (2023), la arquitectura de un microcontrolador se basa en la integración del CPU (unidad central de procesamiento), memoria (RAM y ROM); estos son los periféricos de entrada/salida, donde llegan a ser los convertidores analógico-digitales (ADC), y módulos conocidos como la temporización que está dentro de un solo chip. Esta configuración lo convierte en una solución altamente compacta para sistemas embebidos [50].

Sayan Mitra (2023), muestran cómo los diseños modernos pueden llegar a emplear técnicas de mapeo de memoria unificada para mejorar la eficiencia de acceso, así, reduciendo la necesidad de interfaces externas[50].

Según Liu et al (2023), otra característica clave es la inclusión de modos de bajo consumo, se encuentra sleep, Deep sleep e hibernation, especialmente útiles en sistemas alimentados por baterías. Según Liu et al. Los microcontroladores diseñados para aplicaciones de loT suelen incluir controladores DMA y co-procesadores criptográficos integrados, permitiendo realizar tareas complejas con consumo conocido como energético mínimo [51].



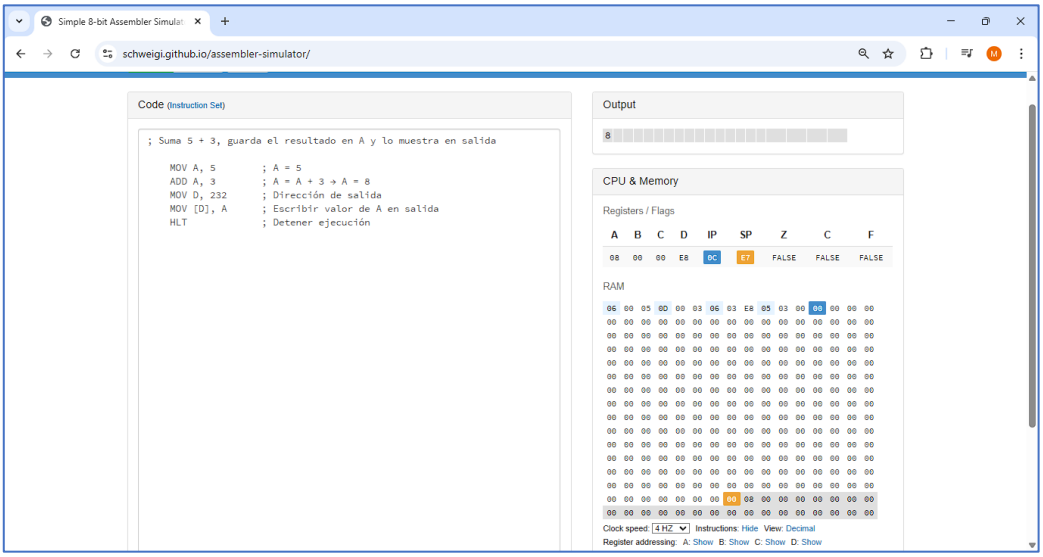
CARACTERISTICAS Y APLICACIONES

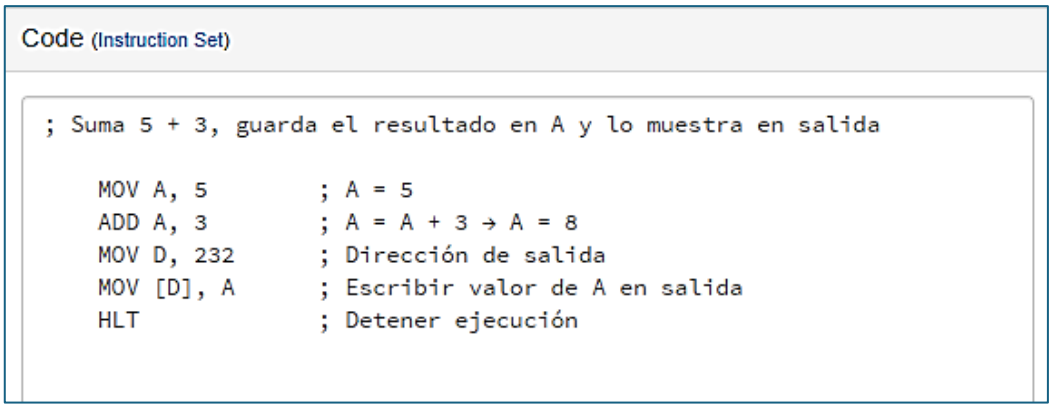
Según Ottaviano et al (2024), las Aplicaciones embebidas criticas necesitan que los microcontroladores operan en modos de ultra-bajo consumo (sleep, Deep sleep) y reaccionen rápidamente ante eventos, manteniendo eficiencia energética. Por ejemplo, el sistema EFM32 de Silicon Labs incorpora modos de suspensión profunda con periféricos autónomos capaces de operar sin despertar la CPU, lo que permite prolongar la vida útil de la batería durante años[52]

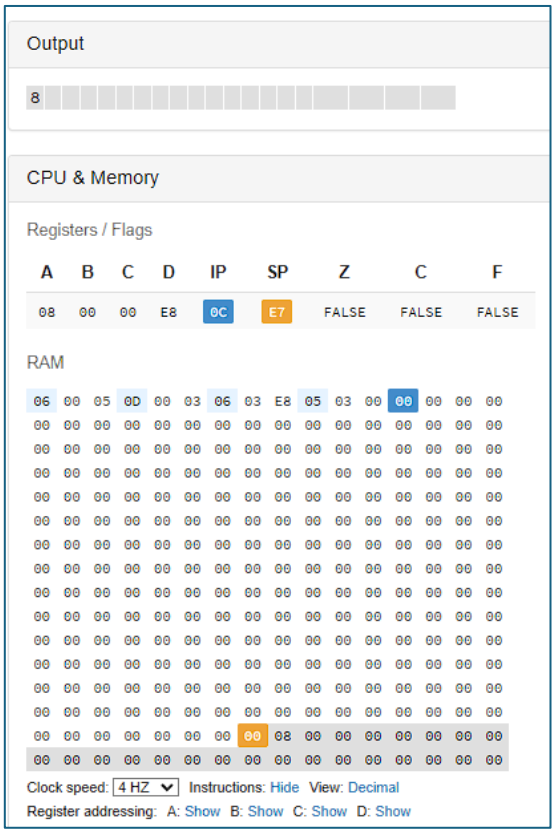
Según Barchi et al (2024), En el ámbito de TinyMl y sensores inteligentes, se ha demostrado el uso de redes neuronales eficientes implementadas en microcontroladores, optimizando la actividad de spikes y reduciendo significativamente el número de ciclos de computación y consumo de energía en tareas de monitoreo estructura [53].

3.4.2 Analizar el ciclo de instrucciones y la velocidad del microprocesador con ejemplos prácticos

3.4.2.1 Ejemplo práctico de una suma







ANÁLISIS DEL CICLO:

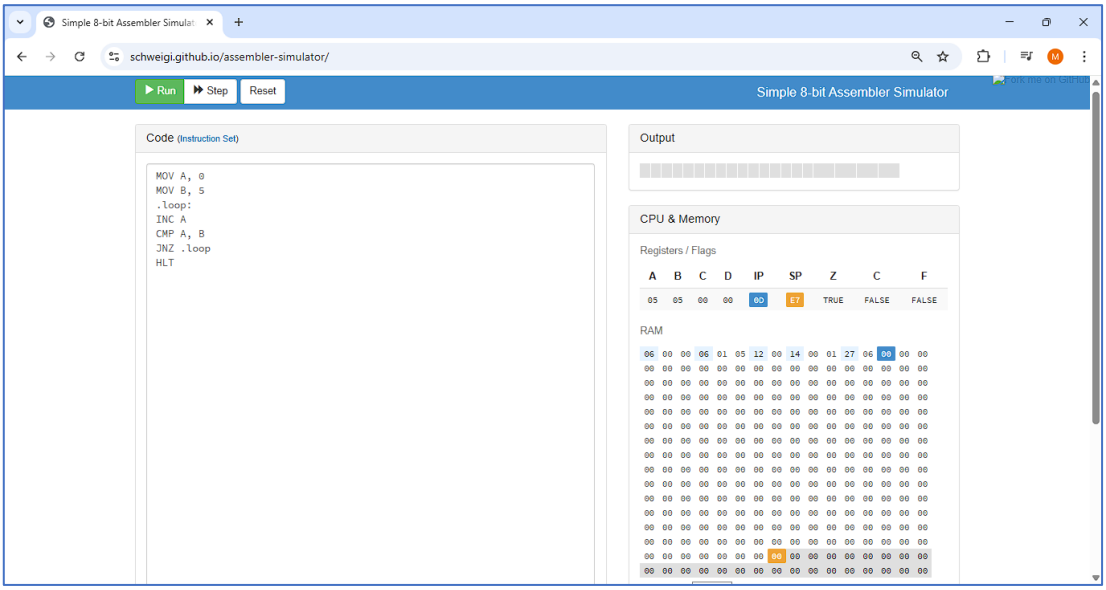
- **MOV** A, 5: 1 ciclo
- **ADD** A, 3: 1 ciclo
- **MOV** D, 232: 1 ciclo
- **MOV** [D], A: 1 ciclo (escritura en salida)
- **HLT**: 1 ciclo
- **Total:** 5 ciclos

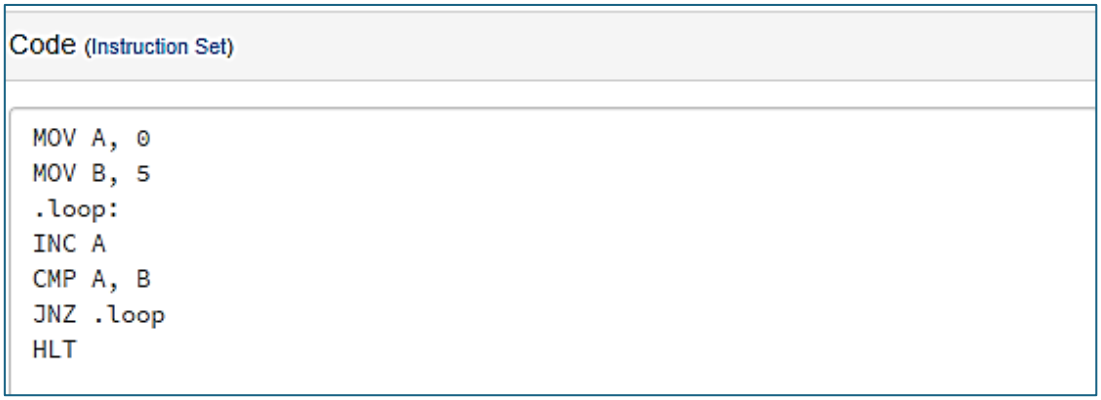
VELOCIDAD:

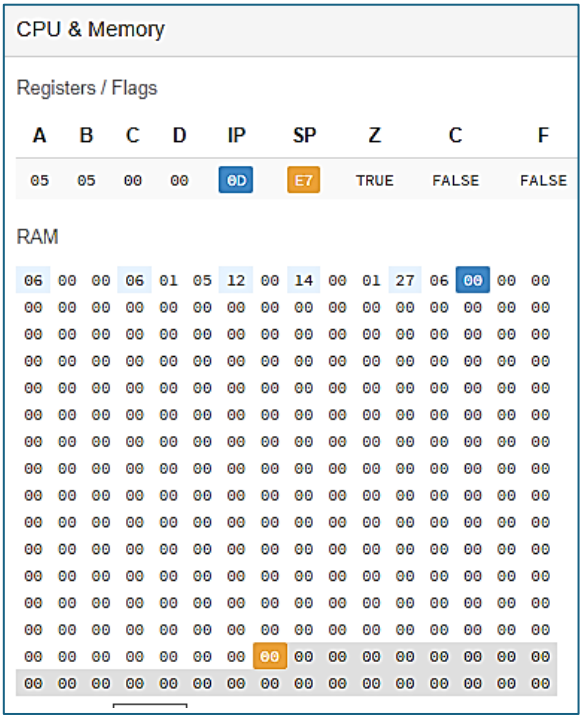
A 1 Hz = 5 segundos

A 4 Hz = 1.25 segundos

3.4.2.2 Ejemplo práctico de bucle que incrementa A hasta 5







CICLO DE INSTRUCCIONES POR VUELTA:

- **INC** A (1 ciclo)
- **CMP** A, B (1 ciclo)
- **JNZ** (1 ciclo)

El bucle se repite 5 veces, entonces:

- 3 ciclos × 5 = 15 ciclos

Más los 2 primeros (MOV A, MOV B) + HLT: 3 ciclos

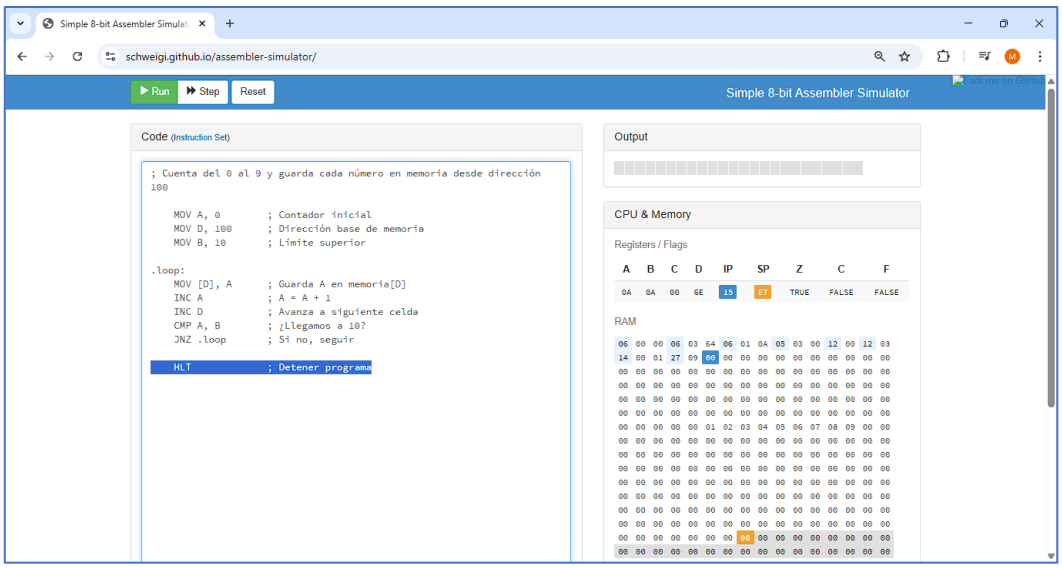
Total: 18 ciclos

VELOCIDAD:

18 ciclos a 1 Hz = 18 segundos

18 ciclos a 4 Hz = 4.5 segundos

3.4.2.3 Ejemplo práctico de contar del 0 al 9 y guardar en memoria



Code (Instruction Set)

```
; Cuenta del 0 al 9 y guarda cada número en memoria desde dirección
100

    MOV A, 0        ; Contador inicial
    MOV D, 100      ; Dirección base de memoria
    MOV B, 10       ; Límite superior

.loop:
    MOV [D], A      ; Guarda A en memoria[D]
    INC A           ; A = A + 1
    INC D           ; Avanza a siguiente celda
    CMP A, B        ; ¿Llegamos a 10?
    JNZ .loop       ; Si no, seguir

    HLT             ; Detener programa
```

CPU & Memory

Registers / Flags

A	B	C	D	IP	SP	Z	C	F
0A	0A	00	6E	15	E7	TRUE	FALSE	FALSE

RAM

06	00	00	06	03	64	06	01	0A	05	03	00	12	00	12	03
14	00	01	27	09	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	01	02	03	04	05	06	07	08	09	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

ANÁLISIS DEL CICLO:

Instrucciones iniciales (MOV A, MOV D, MOV B) : 3 ciclos

Bucle:

- **MOV** [D], A: 1 ciclo
- **INC** A: 1 ciclo
- **INC** D: 1 ciclo
- **CMP** A, B: 1 ciclo
- **JNZ**.loop: 1 ciclo

El bucle se repite 10 veces, entonces:

- =5 × 10 = 50 ciclos

HLT: 1 ciclo

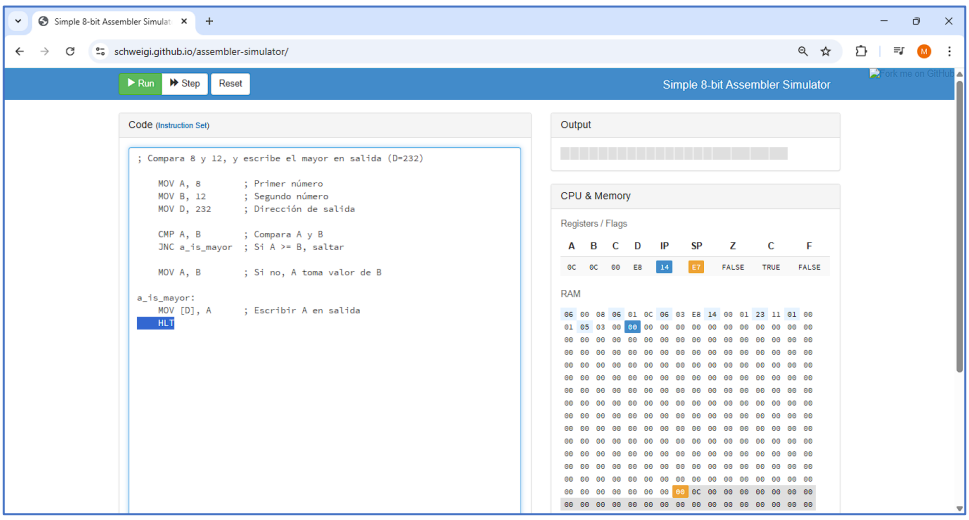
Total: 3 + 50 + 1 = 54 ciclos

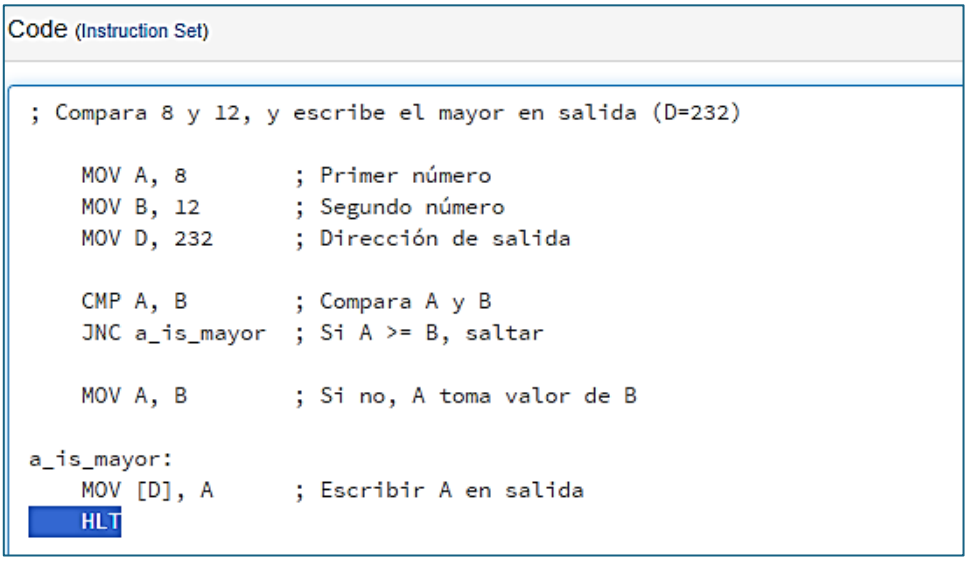
VELOCIDAD:

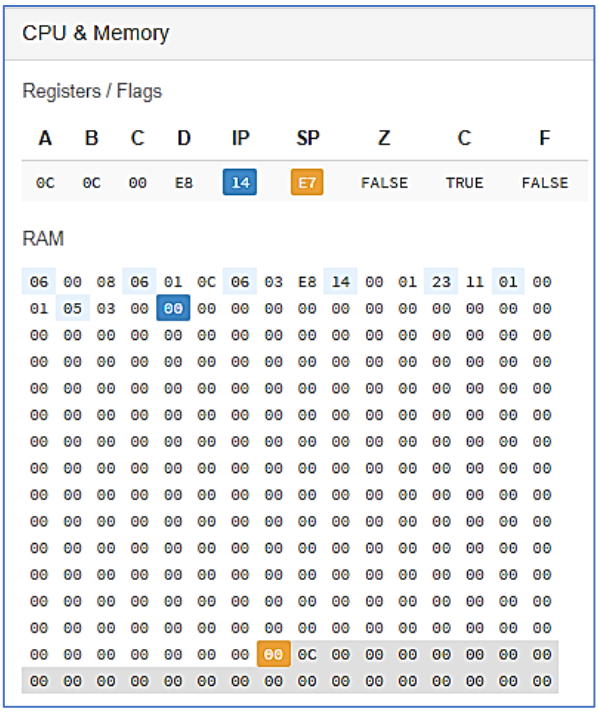
A 1 Hz = 54 segundos

A 4 Hz = 13.5 segundos

3.4.2.4 Ejemplo práctico de comparar 8 y 12, mostrar el mayor







ANÁLISIS DEL CICLO:

MOV A, 8: 1 ciclo

MOV B, 12: 1 ciclo

MOV D, 232: 1 ciclo

CMP A, B: 1 ciclo

JNC: 1 ciclo

En caso de que A < B: se ejecuta MOV A, B: 1 ciclo

MOV [D], A: 1 ciclo

HLT: 1 ciclo

Total: 7 u 8 ciclos (según resultado de comparación)

VELOCIDAD:

A 1 Hz = 7–8 segundos

A 4 Hz = 1.75–2 segundos

4. CONCLUSIÓN

En el estudio de circuitos lógicos se abordaron tanto los circuitos combinacionales como los secuenciales, los cuales constituyen la base del diseño digital al permitir operaciones lógicas y el manejo de información en el tiempo. En cuanto a los microprocesadores y microcontroladores, se analizaron sus características principales, así como sus diferencias en estructura y capacidad de procesamiento.

5. BIBLIOGRAFÍA

[1] Y. Zhang, "Three-Dimensional and Stereoscopic Teaching Reform for Digital Circuit Experiment," 2025, doi: 10.1145/3729605.

[2] G. Donzellini, L. Oneto, D. Ponta, and D. Anguita, *Introduction to digital systems design*. Springer International Publishing, 2018. doi: 10.1007/978-3-319-92804-3.

[3] C. El Helou, P. R. Buskohl, C. E. Tabor, and R. L. Harne, "Digital logic gates in soft, conductive mechanical metamaterials," *Nat Commun*, vol. 12, no. 1, Dec. 2021, doi: 10.1038/s41467-021-21920-y.

[4] X. Lian *et al.*, "Realization of Complete Boolean Logic and Combinational Logic Functionalities on a Memristor-Based Universal Logic Circuit," *Chinese Journal of Electronics*, vol. 33, no. 5, pp. 1137–1146, Sep. 2024, doi: 10.23919/cje.2023.00.091.

[5] P. Ashok and B. Bala Tripura Sundari, "Accuracy Analysis on Design of Stochastic Computing in Arithmetic Components and Combinational Circuit," *Computation*, vol. 11, no. 12, Dec. 2023, doi: 10.3390/computation11120237.

[6] M. Amini-Valashani, M. Ayat, and S. Mirzakuchaki, "Design and analysis of a novel low-power and energy-efficient 18T hybrid full adder," *Microelectronics J*, vol. 74, pp. 49–59, Apr. 2018, doi: 10.1016/J.MEJO.2018.01.018.

[7] Pritty, "Fault correcting adder design for low power applications," *Sci Rep*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-79772-7.

[8] K. Azizbeigi, M. Zamani Pedram, and A. Sanati-Nezhad, "Microfluidic-based processors and circuits design," *Sci Rep*, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-021-90485-z.

[9] M. Vahabi, P. Lyakhov, and A. N. Bahar, "Design and implementation of novel efficient full adder/subtractor circuits based on quantum-dot cellular automata technology," *Applied Sciences (Switzerland)*, vol. 11, no. 18, Sep. 2021, doi: 10.3390/app11188717.

[10] Y. Ye, T. Song, Y. Xie, and C. Li, "Design of All-Optical Subtractors Utilized with Plasmonic Ring Resonators for Optical Computing," *Photonics*, vol. 10, no. 7, Jul. 2023, doi: 10.3390/photonics10070724.

[11] A. Ghadi, “All-optical computing circuits half-subtractor and comparator based on soliton interactions,” *Optik (Stuttg)*, vol. 227, p. 166079, Feb. 2021, doi: 10.1016/J.IJLEO.2020.166079.

[12] S. M. A. Mirizadeh and P. Asghari, “Fault-tolerant quantum reversible full adder/subtractor: Design and implementation,” *Optik (Stuttg)*, vol. 253, p. 168543, Mar. 2022, doi: 10.1016/J.IJLEO.2021.168543.

[13] K. Surendra and S. D. P. Reddy, “High Performance ALU Design using Energy Efficient Borrow Select Subtractor,” vol. Volume 9, Oct. 2022, doi: 10.32628/IJSRST.

[14] F. Parandin *et al.*, “Recent advances in all-optical half-subtractor and full-subtractor based on photonic crystal platforms,” Jun. 01, 2024, *Walter de Gruyter GmbH*. doi: 10.1515/joc-2023-0314.

[15] F. Parandin and N. Bagheri, “Design of a 2 × 1 multiplexer with a ring resonator based on 2D photonic crystals,” *Results in Optics*, vol. 11, May 2023, doi: 10.1016/j.rio.2023.100375.

[16] T. Ghafouri and N. Manavizadeh, “Design and simulation of high-performance 2:1 multiplexer based on side-contacted FED,” *Ain Shams Engineering Journal*, vol. 12, no. 1, pp. 709–716, Mar. 2021, doi: 10.1016/j.asej.2020.05.005.

[17] A. P. Sooriamala, V. S. Solomi, R. Korah, and A. K. Thomas, “Design of Multiplexers using Reversible Logic Technique,” *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, pp. 62–68, Aug. 2023, doi: 10.17762/ijritcc.v11i9s.7397.

[18] A. Kumar, A. Kumar, and A. V. Agrawal, “Field programmable gate array simulation and study on different multiplexer hardware for electronics and communication,” *Computer Science and Information Technologies*, vol. 6, no. 1, pp. 28–39, 2025, doi: 10.11591/csit.v6i1.pp28-39.

[19] D. Xia, Y. Zhang, Y. Tian, M. Xu, and L. Wen, “High-performance and low-power decoder circuits for SRAMs using mixed-logic scheme,” *Integration*, vol. 98, p. 102227, Sep. 2024, doi: 10.1016/J.VLSI.2024.102227.

[20] F. Zahoor *et al.*, “Design implementations of ternary logic systems: A critical review,” Sep. 01, 2024, *Elsevier B.V.* doi: 10.1016/j.rineng.2024.102761.

[21] A. H. Majeed, M. S. Zainal, E. Alkaldy, and D. M. Nor, “Single-Bit Comparator in Quantum-Dot Cellular Automata (QCA) Technology Using Novel QCA-XNOR Gates,” *Journal of Electronic Science and Technology*, vol. 19, no. 3, pp. 263–273, 2021, doi: 10.1016/j.jnlest.2020.100078.

[22] M. Darbandi, S. Seyedi, and H. M. Ridha Al-Khafaji, “An efficient new design of nano-scale comparator circuits using quantum-dot technology,” *Heliyon*, vol. 10, no. 18, Sep. 2024, doi: 10.1016/j.heliyon.2024.e36933.

[23] L. M. Donaire, G. Ortega, E. M. Garzón, and F. Orts, “Lowering the cost of quantum comparator circuits,” *Journal of Supercomputing*, vol. 80, no. 10, pp. 13900–13917, Jul. 2024, doi: 10.1007/s11227-024-05959-4.

[24] A. Saha, N. D. Singh, and D. Pal, “Efficient ternary comparator on CMOS technology,” *Microelectronics J*, vol. 109, p. 105005, Mar. 2021, doi: 10.1016/J.MEJO.2021.105005.

[25] W. Jendernalik, “An Ultra-Low-Energy Analog Comparator for A/D Converters in CMOS Image Sensors,” *Circuits Syst Signal Process*, vol. 36, no. 12, pp. 4829–4843, Dec. 2017, doi: 10.1007/s00034-017-0630-6.

[26] B. Singh Kalyan, B. Singh, and R. Devi, “Data on quantum dot cellular automata based flip flops for designing serial-in-serial-out shift register,” 2024, doi: 10.1016/j.dib.2023.110019.

[27] J. Niu, D. Kim, J. Li, J. Lyu, Y. Lee, and S. Lee, “Reconfigurable Sequential-Logic-in-Memory Implementation Utilizing Ferroelectric Field-Effect Transistors,” *ACS Nano*, vol. 19, p. 5502, 2025, doi: 10.1021/acsnano.4c14062.

[28] M. Alharbi, G. Edwards, and R. Stocker, “Novel ultra-energy-efficient reversible designs of sequential logic quantum-dot cellular automata flip-flop circuits,” *Journal of Supercomputing*, vol. 79, no. 10, pp. 11530–11557, Jul. 2023, doi: 10.1007/s11227-023-05134-1.

[29] Y. K. Maheshwari and M. Sachdev, “VLFF — A very low-power flip-flop with only two clock transistors,” *Integration*, vol. 100, Jan. 2025, doi: 10.1016/j.vlsi.2024.102300.

[30] S. S. Vali and A. kumar N, “Design of low delay low power hybrid logic based flip-flop using FinFET,” *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 9, Sep. 2024, doi: 10.1016/j.prime.2024.100648.

[31] W. Feng, F. Gao, Y. Pugachov, M. Gulitski, and D. Malka, “materials Photonic Crystal Flip-Flops: Recent Developments in All Optical Memory Components,” 2023, doi: 10.3390/ma16196467.

[32] S. Bashir, S. Yaqoob, and S. Ahmed, “Design of QCA based N-bit single layer shift register using efficient JK Flip Flop for nano-communication applications,” *Nano Commun Netw*, vol. 36, p. 100443, Jun. 2023, doi: 10.1016/J.NANCOM.2023.100443.

[33] J.-F. ; Lin, Z.-J. ; Hong, J.-T. ; Wu, X.-Y. ; Tung, C.-H. ; Yang, and Yen, “Citation,” 2022, doi: 10.3390/s22155696.

[34] M. Alharbi, G. Edwards, and R. Stocker, “Novel ultra-energy-efficient reversible designs of sequential logic quantum-dot cellular automata flip-flop circuits,” *Journal of Supercomputing*, vol. 79, no. 10, pp. 11530–11557, Jul. 2023, doi: 10.1007/s11227-023-05134-1.

[35] M. M. Abutaleb, “Robust and efficient quantum-dot cellular automata synchronous counters,” *Microelectronics J*, vol. 61, pp. 6–14, Mar. 2017, doi: 10.1016/J.MEJO.2016.12.013.

[36] M. Gholamnia Roshan and M. Gholami, “Novel level and edge-triggered universal shift registers with low latency in QCA technology,” *Heliyon*, vol. 10, no. 5, p. e26086, Mar. 2024, doi: 10.1016/J.HELIYON.2024.E26086.

[37] G. Mohmed, A. Lotfi, and A. Pourabdollah, “Enhanced fuzzy finite state machine for human activity modelling and recognition,” *J Ambient Intell Humaniz Comput*, vol. 11, no. 12, pp. 6077–6091, Dec. 2020, doi: 10.1007/s12652-020-01917-z.

[38] P. Darche, “Computation Model and Architecture: Illustration with the von Neumann Approach,” in *Microprocessor 1*, 1st ed., Wiley, 2020, pp. 63–130. doi: 10.1002/9781119779667.ch3.

[39] V. V. Rankovska, “Review of Open-source Microcontroller and Programmable Logic Development Boards,” in *2021 XXX International Scientific Conference Electronics (ET)*, IEEE, Sep. 2021, pp. 1–4. doi: 10.1109/ET52713.2021.9580150.

[40] AMD, “AMD Ryzen™ 7 3700X Controladores y compatibilidad.” Accessed: Jul. 26, 2025. [Online]. Available: https://www.amd.com/es/support/downloads/drivers.html/processors/ryzen/ryzen-3000-series/amd-ryzen-7-3700x.html

[41] Intel, “Procesador Intel® Core™i5-12600K.” Accessed: Jul. 26, 2025. [Online]. Available: https://www.intel.la/content/www/xl/es/products/sku/134589/intel-core-i512600k-processor-20m-cache-up-to-4-90-ghz/specifications.html

[42] Intel, “Procesador Intel® Pentium® 4405U.” Accessed: Jul. 26, 2025. [Online]. Available: https://www.intel.la/content/www/xl/es/products/sku/89611/intel-pentium-processor-4405u-2m-cache-2-10-ghz/specifications.html

[43] Microchip Technology Inc., “PIC16F87XA Data Sheet.” Accessed: Jul. 26, 2025. [Online]. Available: https://www.microchip.com/en-us/product/pic16f877a#Documentation

[44] Microchip Technology Inc., “ATmega328P Data Sheet.” Accessed: Jul. 26, 2025. [Online]. Available: https://www.microchip.com/en-us/product/atmega328p#Documentation

[45] Texas Instruments, “MSP430FR6005 Data Sheet.” Accessed: Jul. 26, 2025. [Online]. Available: https://www.ti.com/es-mx/microcontrollers-mcus-processors/msp430-microcontrollers/products.html

[46] M. Hübner and J. Becker, “Multiprocessor System-on-Chip,” *Multiprocessor System-on-Chip: Hardware Design and Tool Integration*, pp. 1–270, 2011, [Online]. Available: https://link.springer.com/10.1007/978-1-4419-6460-1

[47] A. Calimera, P.-E. Gaillardon, K. Korgaonkar, S. Kvatinsky, and R. Reis, Eds., “VLSI-SoC: Design Trends,” vol. 621, 2021, doi: 10.1007/978-3-030-81641-4.

[48] T. Tang, Y. Man, X. Zhou, and D. Wang, “Pipe-DBT: enhancing dynamic binary translation simulators to support pipeline-level simulation,” *Automated Software Engineering*, vol. 32, no. 2, pp. 1–35, Nov. 2025, doi: 10.1007/S10515-025-00506-8/TABLES/6.

[49] R. J. Colvin and R. C. Su, “Structural Operational Semantics for Functional and Security Verification of Pipelined Processors,” pp. 363–388, 2025, doi: 10.1007/978-3-031-98668-0_18.

[50] “Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023) | ACM Conferences.” Accessed: Jul. 26, 2025. [Online]. Available: https://dl.acm.org/doi/10.1145/3576841

[51] K. Akhunov, E. Yildiz, and K. S. Yildirim, “Enabling Efficient Intermittent Computing on Brand New Microcontrollers via Tracking Programmable Voltage Thresholds,” *ENSsys 2023 - Proceedings of the 2023 11th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems, Part of: SenSys 2023*, vol. 12, pp. 16–22, Nov. 2023, doi: 10.1145/3628353.3628547.

[52] A. Ottaviano *et al.*, “ControlPULP: A RISC-V On-Chip Parallel Power Controller for Many-Core HPC Processors with FPGA-Based Hardware-In-The-Loop Power and Thermal Emulation,” *Int J Parallel Program*, vol. 52, no. 1–2, pp. 93–123, Apr. 2024, doi: 10.1007/S10766-024-00761-4/FIGURES/11.

[53] F. Barchi, E. Parisi, L. Zanatta, A. Bartolini, and A. Acquaviva, “Energy efficient and low-latency spiking neural networks on embedded microcontrollers through spiking activity tuning,” *Neural Comput Appl*, vol. 36, no. 30, pp. 18897–18917, Oct. 2024, doi: 10.1007/S00521-024-10191-5/TABLES/5.

6. ANEXO

https://github.com/Moralito64/GrupoI_Corte2_1/tree/main