Imagen que contiene Icono

Descripción generada automáticamenteLogotipo

Descripción generada automáticamente con confianza media

Instituto Politécnico Nacional

Escuela Superior de Cómputo

Arquitectura de Computadoras

**“Implementación del Microprocesador de 4 bits”**

Alumno:

Malagón Baeza Alan Adrian

Profesor:

Alemán Arce Miguel Ángel

Grupo: 5CV1

**Introducción**

**Objetivo:** Implementar, en lenguaje HDL, el microprocesador de 4 bits de la figura siguiente:

Diagrama

Descripción generada automáticamente

* **Memoria ROM.** Este módulo almacena una instrucción y un dato; así sucesivamente. El contador de programa accederá a cada una de sus direcciones donde la instrucción o el dato pasarán al registro de instrucción o al registro de datos según corresponda. según corresponda.
* **Generador de ciclo de máquina o unidad de control.** Este módulo es la unidad de control del microprocesador y su función es sincronizar y activar la participación de cada uno de los registros internos del microprocesador.
* **Contador de programa.** El contador es un registro interno del microprocesador que proporciona la siguiente dirección de memoria, sea para introducir un dato o una instrucción al microprocesador.
* **Registro de instrucción.** Almacena temporalmente la instrucción que se va a ejecutar en el microprocesador.
* **Decodificador de instrucción**. Es el elemento utilizado para interpretar y ejecutar la instrucción que se requiere realizar en la unidad aritmética y aritmética y lógica.
* **Registro de datos.** Almacena los datos que provienen de la memoria de programa y que se requiere realizar en la unidad aritmética y lógica.
* **Unidad aritmética y lógica (ALU).** Es la parte del microprocesador la parte del microprocesador donde se realizan las operaciones lógicas y aritméticas.
* **Acumulador temporal**. Es el registro que almacena temporalmente el resultado de la última operación realizada dentro de la ALU.
* **Acumulador permanente**. Es el registro que almacena el resultado de última operación realizada por el microprocesador.

**Desarrollo**

* **Programación del generador de ciclo de máquina.**

Este módulo coordina los procesos que realiza el microprocesador; utiliza cinco señales de control que activan en orden secuencial control que activan en orden secuencial los registros internos del microprocesador: registro de instrucción (A), o de instrucción (A), registro de datos (B), registro de datos (B), acumulador temporal (C), acumulador permanente (D) y contador de programa (E). y contador de programa (E). En la siguiente figura se muestra este módulo y el diagrama de tiempo de las señales de activación.

Diagrama

Descripción generada automáticamente con confianza media

El programa quedará de la siguiente manera:

Texto

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Descripción RTL obtenida mediante Vivado 2022.2:

Diagrama

Descripción generada automáticamente

Podemos observar cómo utilizo FlipFlops, multiplexores, operador de suma y bloques de memoria ROM del FPGA seleccionado, y para el control de los datos de salida, se implementó mediante flipflops y bloques de memoria ROM del FPGA seleccionado.

Resultado de la simulación:

Interfaz de usuario gráfica

Descripción generada automáticamente

* **Programación del registro de instrucción (RI).**

Este módulo almacena temporalmente las instrucciones provenientes de la memoria de programa. Su función es guardar el código binario de la instrucción mediante la señal de habilitación (A) que envía el generador de ciclo de máquina.

Imagen que contiene reloj, medidor

Descripción generada automáticamente

El programa quedará de la siguiente manera:

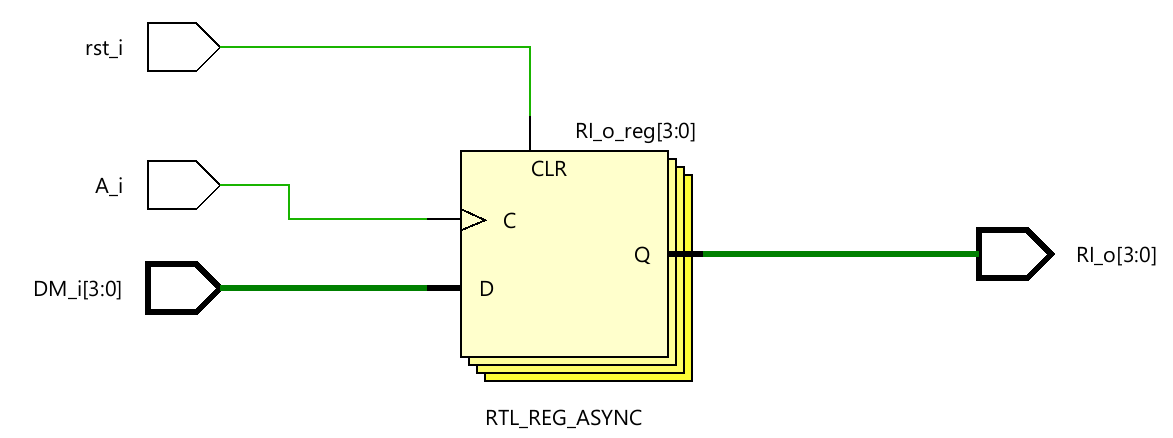
Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Descripción RTL obtenida mediante Vivado 2022.2:



Podemos observar cómo utilizo FlipFlops con su terminal de clk para el control de salida.

Resultado de la simulación:

Interfaz de usuario gráfica

Descripción generada automáticamente

* **Programación del decodificador de instrucción (DI).**

La función de este bloque es convertir el código binario proveniente del registro de instrucción en una acción particular, la cual habilita una de varias operaciones lógicas o aritméticas dentro de la ALU.

Diagrama, Esquemático

Descripción generada automáticamente

El programa quedará de la siguiente manera:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Descripción RTL obtenida mediante Vivado 2022.2:

Gráfico

Descripción generada automáticamente

Podemos observar cómo utilizo un bloque de memoria ROM del FPGA seleccionado para el control de salida.

Resultado de la simulación:

Interfaz de usuario gráfica

Descripción generada automáticamente

* **Programación del registro de datos (RD).**

Este módulo almacena temporalmente los datos provenientes de la memoria de programa. Su función es guardar el dato correspondiente mediante la señal de habilitación (B) que envía el generador de ciclo de máquina.

Diagrama

Descripción generada automáticamente

El programa quedará de la siguiente manera:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Descripción RTL obtenida mediante Vivado 2022.2:

Diagrama, Esquemático

Descripción generada automáticamente

Podemos observar cómo utilizo FlipFlops con su terminal de clk para el control de salida.

Resultado de la simulación:

Interfaz de usuario gráfica

Descripción generada automáticamente

* **Programación del contador de programa (PC).**

Sin duda un microprocesador requiere periféricos externos que le auxilien en sus periféricos externos que le auxilien en su funcionamiento. En nuestro ejemplo, el contador de miento. En nuestro ejemplo, el contador de programa es un elemento que genera el bus de direcciones, ya sea para direccionar una memoria de programa (ROM), una memoria de datos (RAM) o ambas.

**Diagrama

Descripción generada automáticamente**

Como puede observarse, el contador se incrementa cada que se genera la señal (E) proveniente del generador ciclo de máquina.

El programa quedará de la siguiente manera:

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

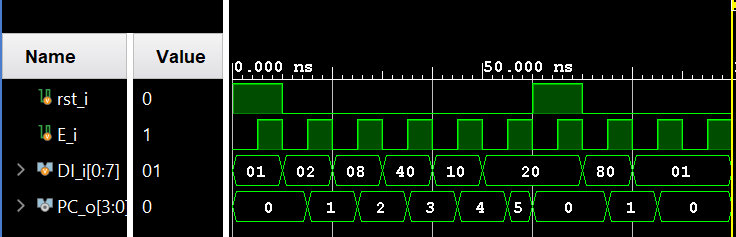
Descripción RTL obtenida mediante Vivado 2022.2:

Gráfico

Descripción generada automáticamente

Podemos observar cómo utilizó un operador de suma y un multiplexor, y para el control del dato a la salida, se implementó mediante flipflops con su terminal de clk.

Resultado de la simulación:



* **Programación de la unidad aritmética y lógica (ALU).**

Iniciaremos por la definición de las instrucciones que va a ejecutar. Este será nuestro set de instrucciones para el pequeño procesador de 4 bits. (ISA, Instruction Set Architechture)

Tabla

Descripción generada automáticamente

Tal como su nombre indica, la función nombre indica, la función de este bloque es realizar las operaciones aritméticas y lógicas del microprocesador. Según se aprecia en la figura, la ALU de nuestro ejemplo puede llevar a cabo ocho operaciones, las cuales se refieren por medio de un código de operación de cuatro bits (tabla).

Diagrama

Descripción generada automáticamente

Cabe mencionar que seis de las ocho operaciones que realiza la ALU requieren dos datos para funcionar: uno se almacena con anterioridad en el acumulador y el otro proviene del registro de datos. Observe que la única operación que no requiere dos datos es la función de "invertir", ya que se realiza invirtiendo el contenido del acumulador. Note también que a la salida de cada bloque de operación (and, or, xor, etc.) se encuentra un buffer triestado activo en alto que, con ayuda del decodificador de instrucción, habilita una de las siete salidas correspondientes a cada operación. Observe que el resultado de las operaciones se almacena de nuevo en el acumulador. En este caso utilizamos este caso utilizamos una señal auxiliar llamada OP (operación), la cual guardará temporalmente el resultado de dicha operación y luego lo canalizará al acumulador temporal.

El programa quedará de la siguiente manera:

Interfaz de usuario gráfica, Texto, Correo electrónico

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Descripción RTL obtenida mediante Vivado 2022.2:

Diagrama

Descripción generada automáticamente

Podemos observar cómo utilizo multiplexores, operador inversor, de suma, xor, or y and, y para el control del dato a la salida, se implementó mediante un multiplexor.

Resultado de la simulación:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

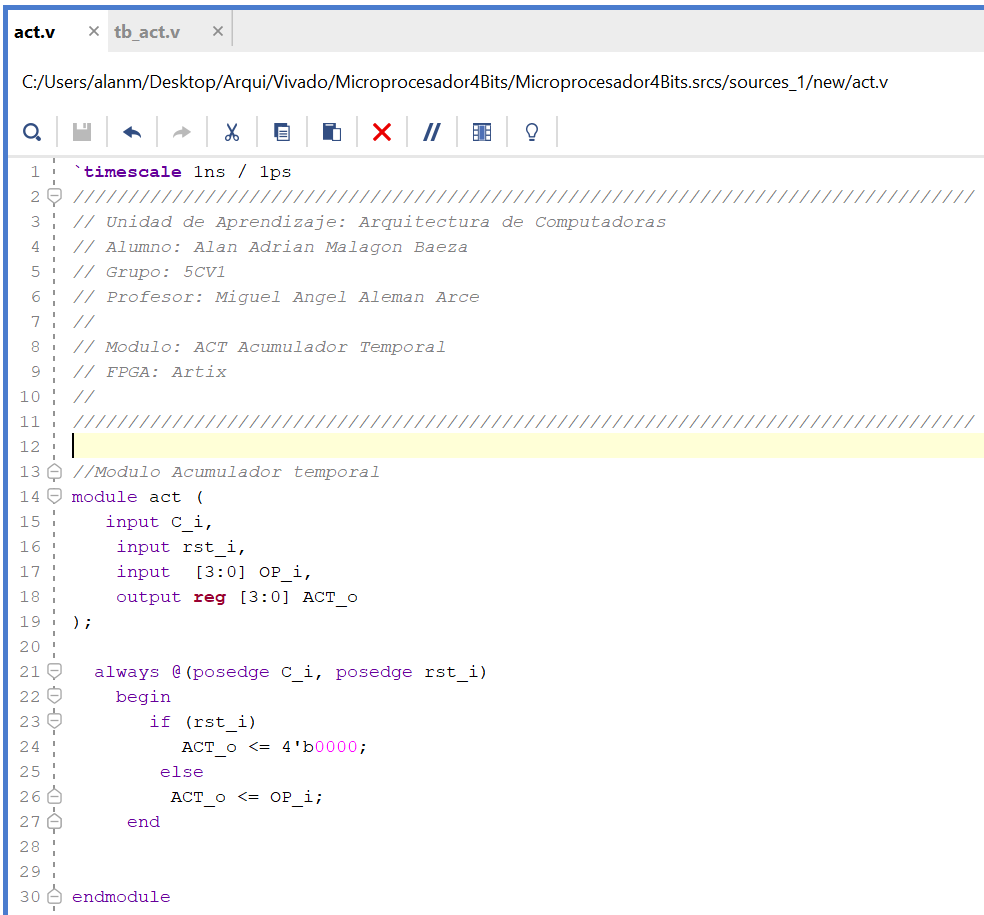
* **Programación del acumulador temporal (ACT).**

La función de este módulo es almacenar temporalmente el resultado proveniente de la ALU (OP) y después canalizarlo por medio de su salida (ACT) al acumulador permanente; este dato se almacena mediante su señal de habilitación (C) proveniente del generador ciclo de máquina.

Diagrama

Descripción generada automáticamente

El programa quedará de la siguiente manera:



Texto

Descripción generada automáticamente

Descripción RTL obtenida mediante Vivado 2022.2:

Diagrama

Descripción generada automáticamente

Podemos observar flipflops con su terminal de clk para el control del dato a la salida.

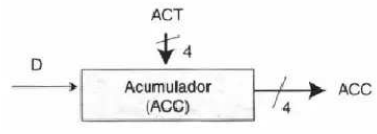
Resultado de la simulación:

Interfaz de usuario gráfica

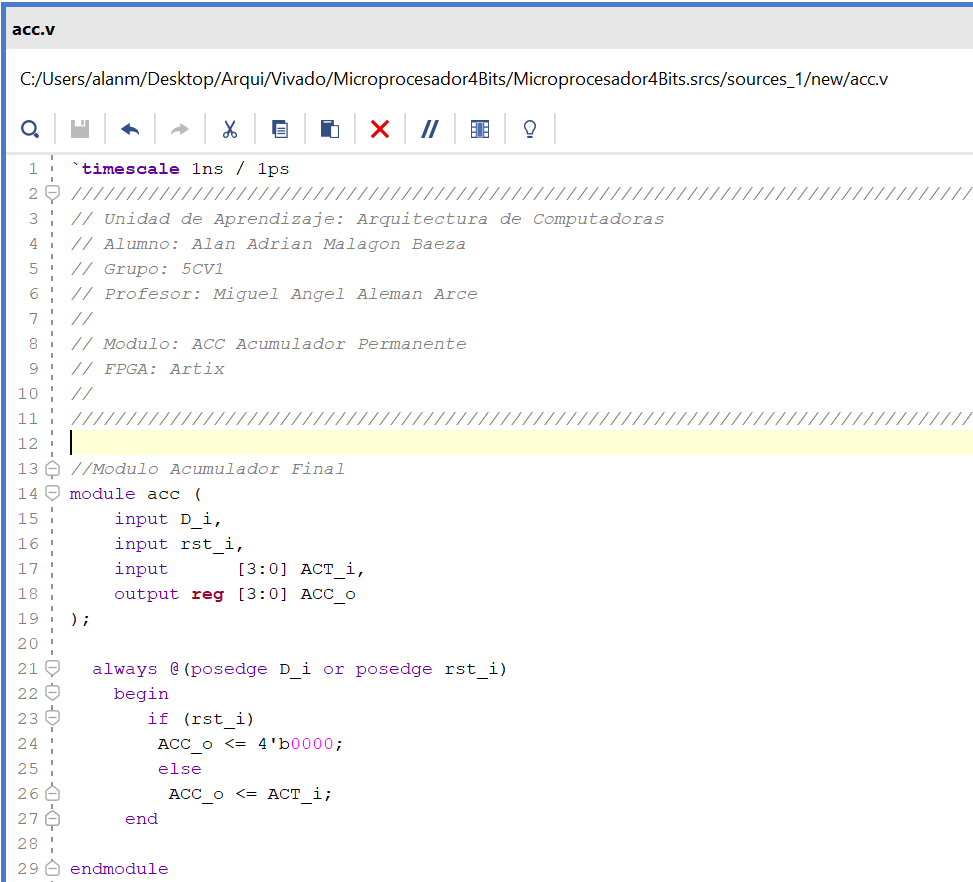
Descripción generada automáticamente

* **Programación del acumulador permanente (ACC).**

La función de este módulo es almacenar el resultado final de la última operación realizada por la ALU, ya sea para enviarlo como aplicación externa o retroalimentar al microprocesador. Este dato se almacena mediante su señal de habilitación (D) proveniente del generador de ciclo de máquina.

****

El programa quedará de la siguiente manera:



Texto

Descripción generada automáticamente

Descripción RTL obtenida mediante Vivado 2022.2:

Diagrama

Descripción generada automáticamente

Podemos observar flipflops con su terminal de clk para el control del dato a la salida.

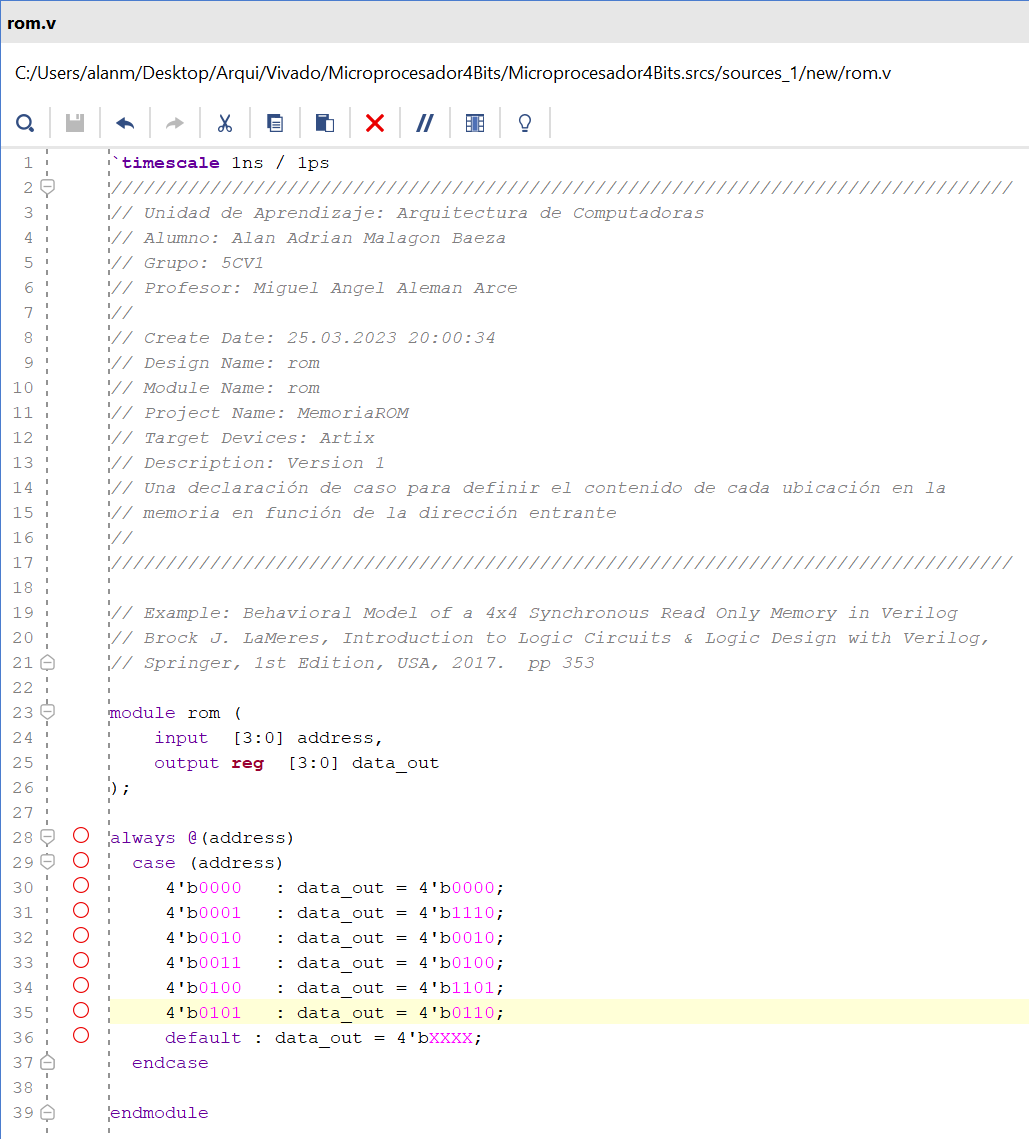
Resultado de la simulación:

**Interfaz de usuario gráfica

Descripción generada automáticamente**

* **Programación de la memoria de programa ROM**

El programa quedará de la siguiente manera:

****

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Descripción RTL obtenida mediante Vivado 2022.2:

**Gráfico

Descripción generada automáticamente**

Como se puede observar se utilizó un bloque de memoria ROM del FPGA seleccionado.

Resultado de la simulación:

Interfaz de usuario gráfica

Descripción generada automáticamente

Como se observa en la simulación, los resultados son los almacenados en la memoria de acuerdo con la localidad.

**Programación del microprocesador de 4 bits**

El programa quedará de la siguiente manera:

Tabla

Descripción generada automáticamente

Texto

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente con confianza media

Descripción RTL obtenida mediante Vivado 2022.2:

Diagrama, Esquemático

Descripción generada automáticamente

Podemos como utilizó todos los módulos anteriormente descritos (GCM, DI, PC, RD, RI, ACT, ACC y ALU) con su terminal de clk y rst.

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

Resultado de la simulación:

**Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente**

**Programación del microprocesador de 4 bits con ROM**

El programa quedará de la siguiente manera:

Tabla

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente Tabla

Descripción generada automáticamente

Se agregaron nuevas operaciones para la ALU y el DI : NOR, NAND XNOR, MAYOR QUE, MENOR QUE, IGUAL, Y RESTA.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Descripción RTL obtenida mediante Vivado 2022.2:

Diagrama, Esquemático

Descripción generada automáticamente

Podemos como utilizó todos los módulos anteriormente descritos (GCM, DI, PC, ROM, RD, RI, ACT, ACC y ALU) con su terminal de clk y rst.

Diagrama

Descripción generada automáticamente con confianza baja

Resultado de la simulación:

Interfaz de usuario gráfica

Descripción generada automáticamente

DI=03 -> MENOR QUE  
ACC=4  
RD=6  
OP = (4<6)? 1:0=1

Cuando D está en flanco de subida ACT=ACC

Cuando C está en flanco de subida ACT=OP

Cuando B esta en flanco de subida RD=DM

Cuando  
Suma: RI=0 -> DI=80  
XOR: RI=2 -> DI=20  
MENOR: RI=d -> DI= 3

Cuando A está en flanco de subida RI=DM

Cuando   
PC=0 -> DM=0  
PC=1 -> DM=e  
PC=2 -> DM=2  
PC=3 -> DM=4  
PC=4 -> DM=d  
PC=5 -> DM=6  
PC=6 y 7 -> DM=X  
porque no están asignados en rom

**Conclusión**

En conclusión, un microprocesador de 4 bits es un tipo de procesador que se utiliza para realizar tareas relativamente simples y que requieren menos recursos de procesamiento en comparación con los procesadores más modernos y avanzados. Aunque puede no ser tan potente como los microprocesadores de 8 bits o más, todavía tiene su lugar en aplicaciones específicas donde el costo, el tamaño y el consumo de energía son factores críticos.

La arquitectura de un microprocesador de 4 bits generalmente consiste en un conjunto limitado de instrucciones y registros, lo que lo hace más fácil de diseñar y fabricar. Además, la arquitectura de 4 bits generalmente requiere menos energía y espacio físico en comparación con procesadores más grandes.

En resumen, la elección de un microprocesador de 4 bits dependerá de las necesidades específicas de una aplicación determinada, teniendo en cuenta factores como la eficiencia energética, el tamaño y el costo.

**Referencia**

1. Brock J. LaMeres, Introduction to Logic Circuits & Logic Design with Verilog, Springer, 1st Edition, USA, 2017.