

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Jeel Guerrero	7-12	Pichardo/micro	27/07/25

Title: RISC-V ISA

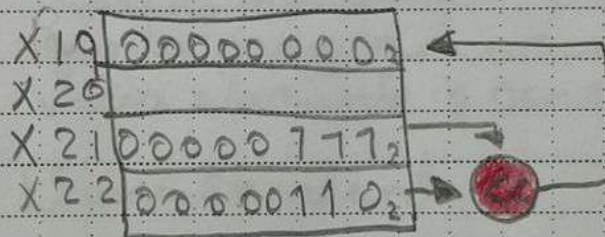
<b>Keyword</b>	<b>Topic:</b> Introducción al conjunto de instrucciones
Microprocesador Hardware procesador Reciente Puertas lógicas VHDL bits	<b>Notes:</b> RISC-V Empezó en el 2010 y actualmente hay muchas universidades y empresas involucradas.  Una de sus ventajas es su diseño modular. El objetivo es aprender a desarrollar un microprocesador más sencillo y el enfoque es más a Hardware que Software.  <b>Questions</b> Sabemos que fue en una universidad en 2010 pero por qué? Como los niveles de abstracción reducen la complejidad?
	Los niveles de abstracción ayudan a reducir la complejidad del diseño en ingeniería y ciencias de la computación.  El diseño se puede realizar en lenguajes de hardware como VHDL.  La implementación final se realiza a través de sintetizadores que convierten el diseño en puertas lógicas.

**Summary:** Introduce al RISC-V como un conjunto de instrucciones abierto modular y eficiente ideal para propósitos educativos e industriales destacando su simplicidad que facilita su desarrollo.



NAME Joel Guerrero	PAGES 2-12	SPEAKER/CLASS Ricardo/micro	DATE - TIME 21/07/25
-----------------------	---------------	--------------------------------	-------------------------

Title: Rise-V ISA

Keyword	Topic: Instrucciones tipo R
Ensamblador Arquitectura de computadores Instrucciones tipo R ISA	<p>Notes:</p> <p>Desplaza a la izquierda el registro RS1 un número RS2 de bits y la guarda en Rd</p> <p>Desplazamiento a la izquierda</p> <p><math>SLL\ rd, rs1, rs2 // rd \leftarrow rs1 \ll rs2</math></p> 
Questions	<p>Cuales la diferencia entre SLT y SLTU?</p> <p>Que ventajas tienen las instrucciones tipo R sobre otras tipos en RISC-V?</p> <p>Permiten optimizar calculos en Hardware, como multiplicaciones y divisiones rapidas</p> <ul style="list-style-type: none"> <li>• Son esenciales en algoritmos de busqueda encriptacion y manipulacion de datos.</li> <li>• Se utilizan en estructuras de control para tomar decisiones en el codigo ensamblador.</li> </ul>

Summary: Hablamos de las instrucciones de tipo R en RISC-V enfocandose en operaciones de comparacion y desplazamiento. Estas instrucciones son fundamentales para manipular datos, calculos eficientes dentro del procesador etc.



NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Jael Guerrero	3-12	Pichardo/Micro	21/07/23

Title: RISC-V ISA

Keyword	Topic: Instrucciones tipo 1
<p>Instrucciones inmediatas</p> <p>Tipo 1</p> <p>ADDI</p> <p>ISA</p> <p>Ensamblador</p>	<p>Notes:</p> <p>Las instrucciones tipo 1 son las que operan entre un registro y una constante. Estas instrucciones son importantes para realizar operaciones aritméticas y lógicas donde una de las operandas es una constante, permitiendo que la programación sea más eficiente y sencilla.</p> <p>Suma Constante</p> <p>Language C <math>\Rightarrow</math> Ensamblador</p> <p><math>i = i + 7;</math>      <code>addi x19, x19, 7</code></p> <p><math>i++</math></p> <p>Language C <math>\Rightarrow</math> Ensamblador</p> <p><math>i = 5;</math>      <code>addi x19, x0, 5</code></p>
<p>Questions</p> <p>En que tipo de programas o algoritmos se usan con mayor frecuencia instrucciones?</p> <p>En que situación es mejor usar ADDI en lugar de ADD?</p>	

Summary: Las instrucciones de tipo 1 en RISC-V que operan entre un registro y un valor inmediato o constante. Estas permiten realizar operaciones aritméticas y lógicas sin necesidad de un 2do registro, optimizando la ejecución.



NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Jael Guerra	4-12	Pichardo/micro	21-07-2025

Title: RISC-V ISA

Keyword	Topic: Instrucciones de carga y almacenamiento
ISA	Notes:
instrucciones tipo S	Las instrucciones de carga y almacenamiento en RISC-V también conocidas como tipo S y tipo B son esenciales para la transferencia de datos entre memoria y registros.
instrucciones tipo B	También hablé de las instrucciones de salto condicional que permite tomar decisiones en la ejecución del código.
Memoria	
Registros	
Questions	
Cómo afectan las instrucciones de almacenamiento al rendimiento del procesador?	tipo S: Almacenamiento de registro a memoria.
Cuál es la diferencia de LB y LBU en RISC-V?	SW Store Word 8 bits Almacena Sh Store Halfword 16 bits Almacena sb Load Byte 32 bits Almacena

Summary: Hablamos de las instrucciones tipo S y B. Estas son fundamentales para la transferencia de datos entre la memoria y los registros, así como para la ejecución de saltos condicionales en la arquitectura RISC-V.



Title: RISC-V ISA

<p><b>Keyword</b></p> <p>ISA</p> <p>Instrucciones de control</p> <p>Flujo del programa</p> <p>memoria</p> <p>Registros</p> <p>Contadores de programa</p> <p><b>Questions</b></p> <p>En que situaciones es mas eficiente usar JALR en vez de JAL?</p> <p>Cual es la relacion entre el PC y las instrucciones de salto?</p>	<p><b>Topic:</b> Direccionalidad de memoria</p> <p><b>Notes:</b></p> <div style="border: 1px solid black; padding: 10px; margin: 10px;"> <p>memoria</p> <p>20 bne x22, x23, offset = 12</p> <p>24 add x19, x20, x21</p> <p>28 beq x0, x0, EXIT</p> <p>32 sub x19, x20, x21</p> <p>EXIT 36</p> <p>40</p> </div> <p>Las desplazamientos relativos (offset) los calcula el ensamblador. Basta con poner etiquetas.</p> <p>Estos saltos relativos los calcula el ensamblador, esto es una gran ventaja por si queremos añadir una instruccion mas. El ensamblador gracias a las etiquetas lo hace facil.</p>
---	--

**Summary:** La recalculation de etiquetas es crucial para mantener la coherencia en los programas. El formato de las instrucciones de control define como se ejecutan los saltos y comparaciones.



NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Jael Guerrero	6-12	Pichardo/micro	21/07/23

Title: RISC-V ISA

Keyword	Topic: Instrucciones de control														
Instrucciones de control	Notes: beq (rs1) (rs2) offset														
Disposición de bits															
Immediate															
comparación de programa															
Contador de registros															
Questions															
Porque es importante entender la disposición de los bits en las instrucciones de control al programar en RISC-V?	<p>en C/2 <math>\rightarrow [2^{13}, 2^{13}-1] = [8192, 8191]</math></p> <p>RISC-V tiene una extensión de instrucciones comprimidas de 16 bits, 2 bytes</p> <p>31 memoria</p> <table border="1"> <tr> <td>20</td> <td>bne x22, x23, offset</td> </tr> <tr> <td>24</td> <td>add x19, x20, x21</td> </tr> <tr> <td>28</td> <td>slli x19, x19, 1</td> </tr> <tr> <td>32</td> <td>beq x0, x0, offset</td> </tr> <tr> <td>36</td> <td>sub x19, x20, x21</td> </tr> <tr> <td>40</td> <td>and x20, x21, x22</td> </tr> <tr> <td>...</td> <td>3   2   1   0</td> </tr> </table> <p>instrucciones de 32 bits</p>	20	bne x22, x23, offset	24	add x19, x20, x21	28	slli x19, x19, 1	32	beq x0, x0, offset	36	sub x19, x20, x21	40	and x20, x21, x22	...	3   2   1   0
20	bne x22, x23, offset														
24	add x19, x20, x21														
28	slli x19, x19, 1														
32	beq x0, x0, offset														
36	sub x19, x20, x21														
40	and x20, x21, x22														
...	3   2   1   0														

Summary: Semuestra como el contador de programa (PC) se ajusta al ejecutar instrucciones de salto. se ve como el programa sigue un flujo diferente dependiendo de la igualdad o desigualdad de valores en los registros comparados.



NAME  
Joel Guerrero

PAGES  
7-12

SPEAKER/CLASS  
Pienardo/muro

DATE - TIME  
21/07/25

Title: RISC-V ISA

### Keyword

Banco de registros  
Registros en RISC-V  
Arquitectura RISC-V  
Instrucciones aritméticas y lógicas

### Questions

Como podria el numero limitado de registros (32 en RISC-V) afectar la eficiencia de programas mas grandes o complejos?

Topic: Banco de registros

### Notes:

0	0	X	0	0	0	0	0	0	0
...									
19	0	X	0	0	0	0	0	0	8
20	0	X	0	0	0	0	0	0	5
21	0	X	0	0	0	0	0	0	2
22	0	X	0	0	0	0	0	0	3
23	0	X	0	0	0	0	0	0	9
24	0	X	0	0	0	0	0	0	7
...									
31									

En el banco de registros tenemos un puerto de entrada que es la direccion del registro y tambien un puerto de Salida que nos dara el valor del registro es data out. La direccion de entrada debe seleccionar a los 32 registros desde el 0 al 31 y para codificarlo necesitaremos 5 bits

### Summary:

El banco de registros es una parte fundamental del procesador. consiste en un conjunto de registros que se utilizan para almacenar datos de forma temporal durante la ejecucion del programa.



NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Jael Guerrero	8-12	Pienardo/micro	21/07/25

Title: RISC-V ISA

Keyword	Topic: Instrucciones tipo R																						
RISC-V	Notes:																						
instrucciones	Son aquellas que operan exclusivamente con registros y no utilizan valores inmediatos ni direcciones de memoria.																						
tipo R	Su formato de 32 bits incluye campos específicos para: registros fuente, el registro destino y el código de operación.																						
formato de instrucciones																							
opcode																							
Registros																							
SLH																							
SRL																							
Questions																							
¿cómo se realiza una suma entre dos registros en RISC-V?																							
¿qué ventajas tiene el tipo R en comparación con los otros tipos de instrucciones en RISC-V?																							
	<table> <thead> <tr> <th></th><th>Tipo R</th></tr> </thead> <tbody> <tr> <td>Add</td><td>add</td></tr> <tr> <td>Subtract</td><td>sub</td></tr> <tr> <td>AND</td><td>and</td></tr> <tr> <td>OR</td><td>or</td></tr> <tr> <td>Exclusive OR</td><td>xor</td></tr> <tr> <td>Set if less than</td><td>slt</td></tr> <tr> <td>Set if less than unsigned</td><td>sltu</td></tr> <tr> <td>Shift left logic</td><td>sll</td></tr> <tr> <td>Shift Right logic</td><td>srl</td></tr> <tr> <td>Shift Right Arithmetic</td><td>sra</td></tr> </tbody> </table>		Tipo R	Add	add	Subtract	sub	AND	and	OR	or	Exclusive OR	xor	Set if less than	slt	Set if less than unsigned	sltu	Shift left logic	sll	Shift Right logic	srl	Shift Right Arithmetic	sra
	Tipo R																						
Add	add																						
Subtract	sub																						
AND	and																						
OR	or																						
Exclusive OR	xor																						
Set if less than	slt																						
Set if less than unsigned	sltu																						
Shift left logic	sll																						
Shift Right logic	srl																						
Shift Right Arithmetic	sra																						

Summary: Son aquellas que operan exclusivamente con registros, sin usar valores inmediatos ni direcciones de memoria. Su formato de 32 bits incluye campos para los registros fuente (rs1, rs2), el registro destino (rd), etc.

By Carlos Pichardo Vinque



NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Jael Guerrero	9-12	Pienardo/micro	29/07/25

Title: RISC-V ISA

Keyword	Topic: Diseño básico
ALU Tipo R Tipo I multiplicar manejo de constantes Registra Señales de control	Notes:  Tipo R: Se utilizan para operaciones entre registros, donde ambos operandos provienen del banco de registros.  Tipo I: Se emplean para operaciones que involucran un registro y un valor inmediato, destacando la diferencia en sus campos, especialmente el uso del inmediato.  Se necesita de un multiplexor que seleccione entre el segundo operando de un registro para instrucciones de tipo I o el valor inmediato antes de ingresar a la ALU.  Señales de control generan señales para seleccionar la operación adecuada (Suma, resta, etc.).
Questions Porque es importante que el diseño de la ALU sea compatible con multiplexores formados de instrucciones? Para que se utilice el valor inmediato en las instrucciones Tipo I?	

Summary: Se detalla como las instrucciones tipo I manejan valores inmediatos, la importancia de un multiplexor para seleccionar entre un operando de registro o inmediato, como la unidad de control genera señales para ejecutar correctamente cada instrucción.



Title: RISC-V ISA

Keyword

Banca de registros

Load Word

Formato

Tipo I

ALU

acceso a memoria

Dirección de memoria

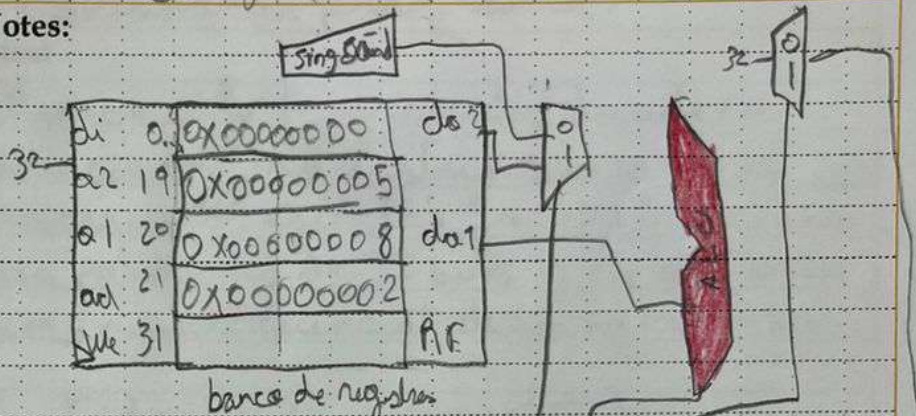
Questions

Que sucedería si la conexión entre la salida de la ALU y la unidad de memoria fallara?

Que papel juega la ALU en la instrucción load word?

Topic: Distinta paraca e instrucciones de carga desde la memoria.

Notes:



Implementación de Hardware

Para la instrucción load word.

Se agregan modulos claves como

la unidad de memoria, conectando

la salida de la ALU a la

dirección de memoria y la

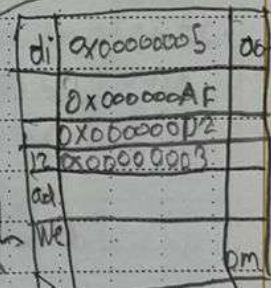
salida de datos de memoria

en el banco de registros. También

se detallan las señales de control

necesarias para habilitar la lectura

de memoria y la escritura en registros.



Summary:

Al implementar una instrucción load word, se revisa su formato de instrucción (Tipo I), el acceso a memoria sumando un inmediato al registro fuente, y la escritura del dato en el registro de destino. También habla de la implementación del Hardware; conexión de la ALU etc.



NAME Joel Guerrero	PAGES 11-12	SPEAKER/CLASS Pichardo/micro	DATE - TIME 29/07/23
-----------------------	----------------	---------------------------------	-------------------------

Title: RISC-V ISA

Keyword ALU Arquitectura de procesadores Ruta de datos Unidad de control Banco de registros RISC-V	Topic: memoria de programa y el Contador de programa.
Questions Por que el PC se incrementa en 4 y no en otro valor? Por que estas instrucciones del RISC-V son de 32 bits?	Notes: Memoria de Programa: es donde se almacenan las instrucciones que el procesador ejecuta, las instrucciones son palabras de 32 bits y se obtienen de manera secuencial desde la memoria. Contador de programa (PC): Es un registro que guarda la direccion de la instruccion actual que se esta ejecutando. Despues de cada instruccion, el PC se incrementa en 4 para apuntar la siguiente instruccion. Conexiones de la ruta de datos: Las instrucciones se conecta a diferentes componentes del procesador, como el banco de registros, la ALU y a unidad de control.

Summary: La memoria del programa almacena instrucciones de 32 bits que se obtienen secuencialmente mientras que el PC guarda la direccion de la instruccion actual y se incrementa en 4 despues de cada ejecucion para determinar la operacion.



NAME	PAGES	SPEAKER/CLASS	DATE - TIME
José Guerra	12-12	Pichardo/mier	29/07/23

Title: TRISC-V ISA

Keyword	Topic: Instrucciones de control de flujo (branching)
Control de flujo Branches ALU Controlador de Programa Señales de control	Notes: beq: Salta si dos registros son iguales bne: Salta si dos registros no son iguales blt: Salta si un registro es menor que otro. bge: Salta si un registro es mayor o igual que otro. bltu: Salta si un registro es menor que otro. bgeu: Salta si un registro es mayor o igual que otro. Para ejecutar un branch, el procesador debe comparar los valores de dos registros y determinar si se cumple la condición. (por ejemplo si un registro es mayor que otro). Esto requiere lógica adicional para evaluar condiciones como igualdad, desigualdad, mayor que, menor que, etc.
Questions	
¿Por qué son importantes las instrucciones de branch en la programación y la ejecución de código?	

Summary: Los Branches en un diseño básico de un Procesador RISC-V. Estas instrucciones permiten al procesador cambiar el flujo de ejecución del programa en función de condiciones específicas, lo que es esencial para implementar bucles, condicionales y saltos en el código.