

Title: Risc-v

| | |
|---|--|
| Keyword Risc-v Registros codigo de maquina niveles de abstraccion principios de diseno | Topic: Introduccion |
| Questions | Notes: Risc-v es una arquitectura de computador de instrucciones (ISA) abierta y modular. Se divide como los procesadores ejecutan instrucciones de manera eficiente utilizando registros rápidos y código máquina. Los niveles de abstracción organizan el diseño desde el hardware físico hasta el software, minimizando los principios de diseño como simplicidad y modularidad para una eficiencia. Los operadores de almacenamiento en registros (32 en Risc-v) o memoria, y las operaciones aritméticas básicas incluyen suma, resta y comparaciones. Su diseño optimiza la velocidad y facilita su implementación en diversos dispositivos. |

Summary: Risc-v es una ISA abierta que prioriza simplicidad y eficiencia en los procesadores. Usa niveles de abstracción para separar hardware y software con operando gestionados en registros rápidos y código máquina eficiente. Sus principios de diseño incluyen modularidad y uniformidad.

Title:

| Keyword | Topic: |
|--|--|
| <p>operandos</p> <p>registros</p> <p>codigos</p> <p>operadores</p> <p>AND</p> <p>OR</p> <p>XOR</p> | <p>Notes: las instrucciones tipo R en RISC-V operando almacenado en registro (32 bits) como (X1, X2) y realizan operaciones aritmeticas y logicas. Instruccion AND, OR, XOR. Su codigo maquina tiene 32 bits y campos como opcode (tipo de operacion), rs1 y rs2 (registros fuente) y rd (registro destino). Estas instrucciones no interactuan con la memoria, lo que lo hace rapido y eficiente.</p> |
| Questions | <p>Ejemplo: add x3, x1, x2 suma los registros x1, x2, guardando el resultado en x3</p> |

Summary: las instrucciones tipo R trabajan entre registros realizando operaciones aritmeticas y logicas. Tienen un formato eficiente de 32 bits en su codigo maquina. Su diseño optimiza la velocidad al evitar el acceso a memoria.

Title:

Compuadores
desplazamiento
SIT
SII
SRI
SRA
registros
código
máquina

Questions

Topic:

Notes:

Los computadores en RISC-V
diseñaron relaciones entre unidades
binarias con instrucciones como SIT (cambio
de la situación sin signo), permitiendo
tomar decisiones condicionales. Los
desplazamientos (SII, SRI, SRA) manipulan los
bits de un registro. Mueven bits a la
izquierda, reemplazando con cero. Multiplica-
ciones rápidas. SRI los mueve a la derecha
con cero (desplazamiento sin signo) y
SRA conserva el signo al desplazar bits
hacia la derecha. Estas operaciones son
rápidas porque trabajan directamente con
registros y se codifican en formato de 32
bits.

Summary:

Los computadores y desplazamiento son esenciales
en RISC-V para tomar decisiones y realizar cálculos
eficientes manipulando bits en registros. con instrucciones
como SIT y SII, optimizan operaciones como
comparaciones, multiplicaciones y divisiones de manera

Title:

| Keyword | Topic: |
|----------------------|--|
| instrucciones tipo 1 | Notes: las instrucciones tipo 1 en RISC-V trabajan con constantes inmediatas y registros. Permiten realizar operaciones aritméticas, lógicas y desplazamiento. La instrucción addi suma una constante a un registro, $\text{Rd} = \text{Rs1} + \text{imm}$. Permiten operaciones lógicas como AND y XOR con valores inmediatos. Para invertir bits, se usa xori con -1. Los desplazamientos slli (desplazamiento a la izquierda) y srai (desplazamiento a la derecha aritmético) manipulan el signo). Estas operaciones son eficientes porque actúan el 160 de memoria externa, manipulando directamente registros. En lenguaje C, se pueden mediante operadores como +, -, ~, y desplazamiento con << y >>. |
| constantes | |
| inmediatos | |
| addi | |
| andi | |
| xori | |
| registros | |
| código máquina | |
| invertir bits | |
| slli | |
| srai | |
| Questions | |

Summary: las instrucciones tipo 1. Permiten realizar operaciones básicas como suma, AND, XOR, desplazamiento y manipulación de bits con valores inmediatos y registros. Se implementan en RISC-V y lenguaje C los hace útiles para control eficiente y control directo del hardware.

Title:

Keyword
Carga y Almac.

memoria
registros
inmediatos
tipo I
tipo R
transferencia de
datos.

Questions

Topic:

Notes: La instrucción de carga y almacenamiento en RISC-V se encarga de mover datos entre la memoria y los registros. Las operaciones de carga (load) como lb (load byte) y lbu (load byte unsigned) cargan datos desde la memoria al registro. Mientras que las de almacenamiento (store), como sb (store byte) transfieren datos de registros a memoria. Instrucciones tipo I trabajan con instrucciones tipo R procesan datos entre registros como add, y xor.

En la transferencia de datos, la instrucción lbu permite crear un byte sin signo desde una dirección de memoria o un registro remitiendo los bits superiores a ceros. Por ejemplo lbu x1, 0(x2) carga byte en la dirección $x2 + 0$ a x1. Las instrucciones de tipo I son clave para estas operaciones porque incluyen la dirección base con un registro.

Summary: Las instrucciones de carga y almacenamiento mueven datos entre memoria y registros, siendo esenciales en operaciones como carga bytes sin signo (lbu). Las instrucciones tipo I trabajan en immediatos para memoria, y las tipo R procesan datos directamente entre registros optimizando transferencia y manipulación de datos.

hugo luis santana

NAME

PAGES

#6

SPEAKER/CLASS

DATE - TIME

22-1-2025

Title:

Keyword

Topic:

Programas almacenados en memoria

Instrucciones de control.

PC Controlador de programas

Memoria principal

Saltos incondicionales

Saltos condicionales

Jal

beq

bne

Direccionamiento M.

Questions

Notes:

El modelo de programa almacenado en memoria consiste en guardar tanto las instrucciones como los datos en la misma memoria principal. Cada instrucción tiene una dirección única, y el controlador de programa (CP) se encarga de apuntar a la próxima instrucción a ejecutar, permitiendo ejecución secuencial o mediante saltos condicionales. Este modelo permite a la CPU modificar el flujo del programa de forma dinámica, esencial para crear estructuras.

Las instrucciones de control manejan el flujo del programa mediante:

Saltos incondicionales

Saltos condicionales

Direccionamiento relativos

Summary:

El modelo de programa almacenado guarda instrucciones y datos en la misma memoria, mientras las instrucciones de control, como jal, beq y bne, manipulan el flujo mediante saltos condicionales e incondicionales usando direccionamiento relativo en desplazamiento.

| | | | |
|----------------------------|-------------|---------------|--------------------------|
| NAME Luis Angel Santana | PAGES #7 | SPEAKER/CLASS | DATE - TIME 22-1-2025 |
|----------------------------|-------------|---------------|--------------------------|

Title:

| Keyword | Topic: |
|--|--|
| RISC-V Diseño básico Operaciones r. Niveles de abstracción Arquitectura Código máquina Banko registro X0 a X31 Propósito general | Notes: En RISC-V, el diseño básico se centra en operaciones básicas registradas, clave para el procesamiento de datos en la CPU. Las operaciones se dividen en dos niveles de abstracción: Arquitectura (instrucciones y registros visibles al programador) y Microarquitectura (implementación interna de la CPU y código de máquina representado por bitstream de las instrucciones). |
| Questions | <p>El banco de registro contiene 32 registros de propósito general (X0 a X31), cada uno con 32 bits. X0 siempre tiene el valor 0 y se utiliza para operaciones lógicas entre registros (add, sub, and, or, xor). Por ejemplo, add, X1, X2, X3 almacenará en X1 la suma de X2 + X3.</p> <p>Este diseño asegura que todas las operaciones entre registros sean rápidas y uniformes, eliminando la necesidad de acceder a memoria externa lo que optimiza el rendimiento.</p> |

Summary: RISC-V organiza sus operaciones en niveles de abstracción, destacando las operaciones entre registros. Mediante un banco de 32 registros, instrucciones tipo R como add, sub, x xor son esenciales para realizar cálculos rápidos sin depender de memoria, optimizando el rendimiento del procesador.

Title:

Keyword
 Carga base
 Memoria.
 Transferencia de
 datos
 Banco registros
 Memoria principal
 lw (load word)
 sw (store word)
 lh (load halfword)
 sh (store halfword)

Topic:

Notes: En RISC-U a diferencia de otros arquitecturas las instrucciones son estrictamente para transferir datos entre la memoria y transferir datos entre registros y memoria principal a base de registro. ejemplo: lwxi offset(xz) carga una palabra de 32 bits desde la direccion calculada como $base + offset \times 4$ la almacenara en X1. Tambien puede cargar bytes (lb) o datos sin signo (lbu).

Questions

Las instrucciones de transferencia de datos de un registro a la memoria mediante sw (store word) o sus variantes como sh (store halfword) y sh (store halfword) estas instrucciones son esenciales para mover datos entre registros y direcciones especificas de memoria.

Las operaciones con registros importantes implican trabajar con registros claves como X0, que siempre almacena el valor 0, o X1 para el primer registro o como operandos neutros.

Summary:

El diseño básico 3 de RISC-U cubre las instrucciones de carga y almacenamiento para transferir datos entre memoria y registros. instrucciones como lw y sw garantizan un flujo eficiente de datos, almacenado en registros clave como X0 para operaciones neutras. estas capacidades son esenciales para procesar y manipular datos extendidos CPU.