

NAME

Kennedy Rodríguez H.

PAGES

1/3

SPEAKER/CLASS

Microcontroladores

DATE - TIME

12/9/24

Title Investigación del ISA del ARM M0 con tipos de direccionamiento.

Keyword

ARM

Thumb
Bits

Microcontroladores

Instrucción

Datos

LDR

STR

BEQ

BNE

Questions

¿Qué beneficios
me traería
implementar
el ARM cortex
M0?

Topic: ¿Qué es el ARM cortex-M0 ISA?

El cortex-M0 es compatible con el conjunto de instrucciones ARMv6-M, lo que significa que utiliza instrucciones Thumb de 16 bits (aunque algunos microcontroladores pueden soportar instrucciones de 32 bits). Estas instrucciones aritméticas (suma, resta, multiplicación) y lógicas (AND, OR, XOR, NOT). Son conocidas como operaciones de datos.

- Transferencia de datos: Instrucciones para cargar y almacenar datos desde/hacia la memoria (LDR, STR).

- Control de flujo: Instrucciones de Salto (B, BL), condicionales (BEQ, BNE) y de llamada a subrutinas.

- Manipulación de Bits: Instrucciones específicas para operaciones con bits como rotaciones o máscaras.

Summary: En resumen, el ARM cortex-M0 es un conjunto de instrucciones compacto y eficiente, diseñado específicamente para sistemas embebidos de bajo costo y bajo consumo de energía.

NAME
Kernegay Rodriguez H.

PAGES
2/3

SPEAKER/CLASS
Microcontroladores

DATE - TIME
12/9/24

Title: Investigación del ISA del ARM M0 con tipos de direccionamiento.

Keyword

Operando
Dirección
memoria
Instrucciones
Registros
ARM
Arquitectura

Questions

¿En qué se diferencia este tipo de direccionamiento con los otros?

Topic: Tipos de direccionamiento en ARM cortex-M0

1- Direccionamiento Inmediato: El operando es valor inmediato codificado dentro de la instrucción.
Ejemplo: MOV R0, #5, donde el valor 5 es el operando inmediato.

2- Direccionamiento Directo/Absoluto: La instrucción especifica la dirección de memoria donde se encuentra el operando. Sin embargo, en el ARM cortex-M0, el direccionamiento absoluto no es común, ya que la mayoría de las instrucciones operan con registros.

3- Direccionamiento registrado: El operando está en un registro. Es el más común en ARM debido a la arquitectura Load-Store, donde los cálculos se realizan en registros. Ej: ADD R1, R2, R3, suma los valores en R2 y R3, y coloca el resultado en R1.

Summary: En resumen, vimos los primeros 3 tipos de direccionamiento del ARM, los cuales podemos hacer un símil con C y C# porque se parecen un poco a como son en esos lenguajes la declaración de direcciones y registros.

Title: Investigación del ISA del ARM MO con tipos de direccionamientos

Keyword

Direccionamiento
Desplazamiento
Dirección
memoria
LDR
Almacenamiento
Puntero
Registro

Questions

¿Tendrá el ARM un sistema de instrucciones más completo que los otros?

Topic: Tipos de direccionamiento del ARM

4- Direccionamiento de desplazamiento: Combina un registro base y un desplazamiento inmediato para acceder a una dirección de memoria. Se utiliza mucho en la carga y almacenamiento de datos. Ejemplo: `LDR R0, [R1, #4]`, carga en R0 el valor almacenado en la dirección resultante de sumar 4 al contenido de R1.

5- Direccionamiento Indirecto: Usa el valor de un registro como puntero para acceder a la memoria. Ejemplo: `LDR R0, [R1]`, carga en R0 el valor apuntado por R1.

6- Direccionamiento Indirecto: Similar al de desplazamiento, pero con un desplazamiento que puede ser otro registro. Ejemplo: `LDR R0, [R1, R2]`, suma los valores de R1 y R2 para calcular la dirección de memoria.

7- Direccionamiento pre/post-Indirecto: El pre-Indirecto suma el desplazamiento al registro antes de acceder a la memoria, mientras que el post-Indirecto lo hace después.

Summary: En resumen, aquí terminamos de ver los Tipos de direccionamiento en el ARM. Estos tipos de desplazamiento combinan tipos anteriormente vistos. Básicamente estos tipos de desplazamiento son las formas posibles de desplazar la información.