

Name Bryan Antonio Hernandez	Page 1 de 3 paginas	Speaker microcontroladores	28 1 29
------------------------------------	------------------------	-------------------------------	---------

Title: Assembler para procesador ARM

Keyword	<p>Topic: Estructura y sintaxis del Assembler ARM</p> <p>El Assembler ARM. Utiliza registros como R0, instrucciones como "LDR" y "ADD" para manipular datos, y secciones "Data" y "Text" para organizar el código.</p> <p>-16: Disponible solo en armasm. Ensamblador de origen como instrucciones Thumb de 16 bits.</p> <p>-32: Disponible solo en armasm. Ensamblador de origen como instrucciones ARM de 32 bits.</p> <p>-8: Genera información de depuración</p> <p>-list: Imprime una lista de datos los macros definidos.</p> <p>Thumb: Disponible solo en armasm. Ensamblador de origen como instrucciones Thumb.</p>
---------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Questions	<p>1. ¿Cuáles es el registro mas comun? R0-R7, Estos registros son de proposito general y se utiliza para almacenar datos temporales y resultados de operacion</p> <p>2. ¿Como se realiza una llamada al sistema? Se realiza a travez de la instruccion "SWI" (Software Interrupt), pero es importante saber que esto puede variar entre sistemas operativos y la version de la arquitectura.</p> <p>3. ¿Cuáles son las secciones típicas en un programa ensamblador ARM? Sección "Text" que contiene el código ejecutable, Sección "Data" que almacena datos inicializados y Sección ".bss" que contiene datos no inicializados</p>
-----------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Summary: En el ensamblador ARM, se emplean registros, instrucciones y secciones específicas para organizar y ejecutar el código de bajo nivel. Comprender la estructura y sintaxis es esencial para programar en cualquier arquitectura.

Exerc. 2

APACHE

Nombre: Naya Salazar	Página: 2 de 3 páginas	Speaker: Miguel Hernández	28 1 24
-------------------------	---------------------------	------------------------------	---------

Titulo: Assembler para procesador ESP32 (Xtensa)

Keyword

Topic: Características, claves del Assembler Xtensa

El Assembler Xtensa del ESP32 utiliza registros como "A0", instrucciones específicas para cargar/cargar datos y controlar el flujo, y acciones "call" y "ret".

- **movh=arch:** Especifica la arquitectura de destino, el valor predefinido es 1x6
- **movb=abi:** Especifica la interfaz de llamada de destino, el valor predefinido es 11p32
- **func=opcion7:** Designa la opción específica
- **in=opcion2:** Define la opción específica
- **o=archivo objeto:** Especifica el nombre del archivo de salida. De forma predefinida, el nombre del archivo de objeto.

Question

1. ¿Cuáles son las diferentes clave entre el Assembler Xtensa y el ARM? La sintaxis del ensamblador Xtensa difiere de la ARM, ya que cada arquitectura tiene su propio conjunto de instrucciones.
2. ¿Cómo se gestionan las llamadas al sistema en el ESP32? Las llamadas al sistema pueden variar según el entorno y el sistema operativo utilizado.
3. ¿Cuáles son los registros más usados?
 - "A0": Utilizado para pasar argumentos a una función
 - "A1", "A2", etc.: Otros registros de argumentación
 - "ACC0", "ACC1": Registros de acumulador para operaciones.

Summary: El Assembler del ESP32 (Xtensa) se distingue por sus características únicas, ofreciendo una amplitud en los registros. Su arquitectura Xtensa ofrece una variedad de registros, como "A0", que se utiliza de manera específica para cierto propósito proporcionando flexibilidad y eficiencia.

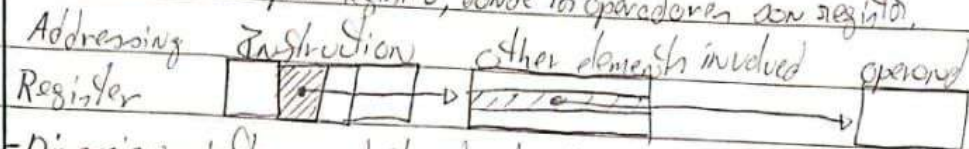
APACHE

Title: Modos de direccionamiento en MIPS

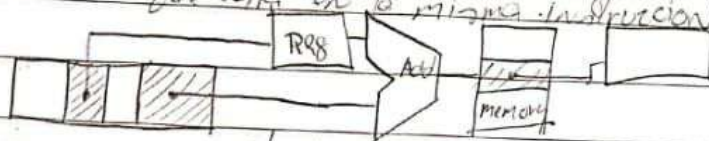
Keyword

Topic: Diferentes modos de direccionamiento

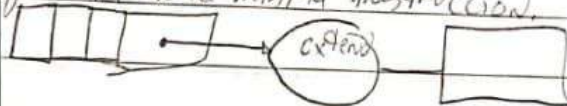
Direccionamiento por registro, donde los operandos son registros.



- Direccionamiento inmediato, donde los operandos es una constante que está en la misma instrucción



- Direccionamiento inmediato, donde uno de los operandos es una constante que está en la misma instrucción.



Question

1. ¿Cómo difieren las instrucciones de direccionamiento inmediato de las direccionamiento inmediato? Direccionamiento inmediato es una constante que se encuentra en la propia instrucción.
2. ¿Cuándo es útil utilizar el direccionamiento relativo al PC? se utiliza para formar una dirección sumando una constante en la instrucción con el registro "PC". Es útil para instrucciones de salto condicional donde la dirección de salto depende de una condición.
3. ¿Qué beneficios aporta el direccionamiento base o desplazamiento a la ejecución de instrucciones?

Summary: En esta arquitectura, los modos de direccionamiento abarcan desde el uso de registros hasta operandos con direccionamiento de memoria, facilitando eficiente de instrucciones. Los modos, como el inmediato, base o desplazamiento de operandos proporcionan flexibilidad y optimización en la manipulación de operandos.