

Directivas: También se les conoce como pseudo operaciones, estas son comandos que afectan al lenguaje ensamblador, mas no al microprocesador pues no generan código ejecutable, sirven para controlar la manera en que se “ensambla” un programa.

Directivas utilizadas para cualquier programa

THUMB: Le indica a Ensamblador que tipo de instrucciones vamos a utilizar. Las instrucciones thumb son un juego de instrucciones reducido, de 16 bits. Esto ayuda a que se necesite la mitad de la memoria al programar.

AREA: Nos permite definir bloques de datos y código. Esto permite asignar datos a la memoria Ram y código en la memoria Rom.

|text.| Nos indica que a partir de allí se empezará a escribir el código.

Code: Esta corresponde al código de maquina.

Readonly: Nos indica que el área estará ubicada en la ROM

ALIGN: Alinea la sección en un limite de 2^n bytes

END: Nos indica que el programa ha terminado

Directivas para asignación

EQU: Nos permite asignarle un valor a cualquier variable.

Registros de propósito General

La arquitectura tiene 31 registros de propósito general, de 32 bits cada uno.

De estos solo 16 serán visibles para el usuario, pero no todos son utilizables.

- Puntero de pila: usualmente R13, lleva por siglas SP (Stack Pointer). Sirve para insertar y obtener información de la pila.
- Registro de enlace: R14, con siglas LR (Link Register). Mantiene la dirección de la siguiente instrucción después de un salto en el programa.
- Contador de programa: R15, con siglas PC (Program Counter). Indicador para apuntar a la instrucción dos líneas debajo de la ejecutada en el momento.

Punto flotante

Es una forma de notación científica usada en las computadoras, utilizada para representar números reales de forma mas compacta. Tienden a ser similares a los de registro de propósito general, pero se les cola el prefijo V y el sufijo .F32

Existen sufijos de condicionamiento, los cuales no cambian el significado, pero ayudan a cumplir las condiciones que uno necesite.

EQ= Igual

NE= No Igual

CS Mayor o igual

CC Menor

HI: Mayor

LT Menor con signo.

INSTRUCCIONES DE CARGA y Almacenamiento

LDR Las instrucciones de carga toman valores en memoria y los escriben en registros de propósito general.

MOV: Atribuyen valores inmediatos.

STR: Las instrucciones de almacenamiento toman valores en registros de propósito general y los escriben en memoria.

Instrucciones Aritmeticas

ADD Suma

MUL Multiplicacion

SUB Division

Instrucciones Logicas

AND AND

ORR OR

ORN NOR

Instrucciones para Condicionamiento

B SALTO

BL Salto con enlace

BLX Salto indirecto con enlace

BX Salto indirecto

CBNZ Comparar y saltar si no es igual a cero

CMP comparar

CMN Comparar negativo

BEQ Salto si la comparación anterior es igual.

VADD.F32 Suma {Sd, } Sn, Sm

VCMP.F32 Comparar Sd, Sm Sd, #0.0

VCVT.S32.F32 Convertir de formato punto flotante a entero

VCVT.S32.F32 S2, S3

VMOV.F32 R1, S2

STARTUP

Se compone de:

Declaración de área de pila

▣ Declaración de área de heap (memoria reservada para almacenamiento mientras el programa corre).

▣ Tabla de vectores.

▣ Código de controlador de reinicio.

▣ Código de otros controladores de excepciones.

Por el contenido de este tipo de archivos, su lectura es fundamental para la preparación del dispositivo antes de cargar el programa ya que el Startup indicará cómo se manejarán interrupciones, ubicación de pila, controladores especificados por el usuario y qué sucederá en el programa inmediatamente después de un reinicio o la primera vez que el código se carga en el procesador. Es importante recordar, que este archivo será lo primero en leerse antes de cualquier otro segmento de código y que se ejecuta una sola vez por reinicio o carga, especificando en el control de reinicio el punto exacto de código al que se saltará después en el archivo de código principal, para ejecutar el resto del programa por lo general utilizando alguna instrucción branch.