# **Set de Instrucciones del ARM**

El Set de instrucciones del ARM puede ser dividido en 6 categorías generales

* Instrucciones de Salto.
* Instrucciones de Procesamiento de Datos.
* Instrucciones de Transferencias del registro Status.
* Instrucciones para Carga y Escritura en Memoria.
* Instrucciones para Coprocesador (solo se menciona pero no se desarrollara en el resto del apunte)
* Instrucciones para la Generación de Excepciones.

La mayoría de las instrucciones de procesamiento de datos puede actualizar las cuatro banderas de condiciones en el CPSR ([N]Negativo, [Z]Cero, [C]Acarreo y [V]Desborde) de acuerdo a su resultado

Además todas las instrucciones ARM contiene un campo de 4bit condicional ubicado en el extremo mas significativo, un valor especifico de este campo determina la ejecución incondicional de la instrucción, otros 14 valores condicionan la ejecución de la instrucción, si la las banderas indica que la correspondiente condición es verdadera, entonces la instrucción es ejecutada, caso contrario no ocurre nada y continua a la próxima instrucción.

El nemónico de la condición y su significado se puede ver en la siguiente tabla.

|  |  |  |  |
| --- | --- | --- | --- |
| **Opcode** | **Mnemónico** | **Significado** | **Condición** |
| 0000 | EQ | Igual | Z = 1 |
| 0001 | NE | Distintos | Z = 0 |
| 0010 | CS/HS | Acarreo / Sobre o igual | C = 1 |
| 0011 | CC/LO | Sin Acarreo / Debajo | C = 0 |
| 0100 | MI | Negativo | N = 1 |
| 0101 | PL | Positivo o Cero | N = 0 |
| 0110 | VS | Desborde | V = 1 |
| 0111 | VC | Sin desborde | V = 0 |
| 1000 | HI | Sobre | C = 1 y Z = 0 |
| 1001 | LS | Debajo o Igual | C = 0 o Z = 1 |
| 1010 | GE | Mayor o Igual | N = V |
| 1011 | LT | Menor | N <> V |
| 1100 | GT | Mayor | Z = 0 y N = V |
| 1101 | LE | Menor o Igual | Z = 1 o N <> V |
| 1110 | AL | Sin Condición |  |
| 1111 | No Usado |  | |

Estas 14 condiciones, permiten:

* probar igualdad o desigualdad.
* probar comparaciones <,<=,> y >= en aritmética con o sin signo.
* probar alguna de las 4 banderas individualmente. El decimosexto valor, permite instrucciones alternativas las cuales no admiten ejecución condicional.

## **Instrucciones de Salto**

Así como cualquier instrucción de procesamiento de dato o instrucciones de carga pueden cambiar el flujo del programa simplemente escribiendo el PC, una instrucción estándar de Salto, provee un número de 24bit con signo como corrimiento, permitiendo realizar saltos hacia arriba o abajo dentro de una brecha de +/-32Mb.

Existe también un salto con vinculo o mejor dicho preservando el vinculo, (BL) esta instrucción antes de realizar el salto guarda la posición de la instrucción precedente a la instrucción de salto en R14 o LR, esto provee a la subrutina la posibilidad de retornar desde donde fue llamada copiando simplemente LR a PC.

Las Instrucciones de salto permiten además cambiar de modo de operación ( de ARM a Thumb o de Thumb a ARM ), llamar a subrutinas escritas en ARM desde Thumb o rutinas escritas en Thumb desde ARM.

[Descripción](http://ciii.frc.utn.edu.ar/TecnicasDigitalesII/WebHome/ASMIntroduccionARM/ASMSaltoARM)

## **Instrucciones de Procesamiento de Datos**

Las instrucciones de procesamiento de datos, realizan cálculos en los registros de propósitos generales.

Se dividen en tres categorías:

* Instrucciones Aritméticas / Lógicas.
* Instrucciones de Comparación.
* Instrucciones de Multiplicación.

### **Instrucciones Aritméticas / lógicas**

Realizan operaciones aritméticas y lógicas en uno o dos operandos y escriben el resultado en un registro destino.

Además pueden opcionalmente actualizar los códigos de condición basándose en el resultado.

El primer operador debe ser siempre un registro, el otro operador puede ser:

* un valor inmediato.
* un registro al que se le puede aplicar opcionalmente un corrimiento fijo o a través de otro registro.

En este último caso, se pueden especificar cinco tipos de corrimientos, esto permite que cada instrucción aritmética/lógica pueda realizar además un corrimiento, como resultado, ARM no posee instrucciones de corrimiento dedicadas.

El Contador de Programa (PC) al ser un registro de propósitos generales, las instrucciones aritméticas/lógicas pueden escribir en él un nuevo valor, permitiendo otra via para implementar saltos.

[Descripción](http://ciii.frc.utn.edu.ar/TecnicasDigitalesII/WebHome/ASMIntroduccionARM/ASMAritmeticaARM)

### **Instrucciones de Comparación**

Estas instrucciones poseen el mismo formato que las Aritméticas/Lógicas, realizan también una operación A/L, pero no escriben el resultado en un registro, ellas solo actualizan las banderas de condición, basándose en el resultado.

Los operandos poseen las mismas características de las A/L incluyendo la posibilidad de incorporar un corrimiento a uno de los operadores.

[Descripción](http://ciii.frc.utn.edu.ar/TecnicasDigitalesII/WebHome/ASMIntroduccionARM/ASMAritmeticaARM)

### **Instrucciones de Multiplicación**

Realizan operaciones de multiplicación con enteros.

Podemos distinguir dos tipo de instrucciones de multiplicación

|  |  |  |
| --- | --- | --- |
| Denominación | Tamaño op. | Resultado |
| Normal | 32bits x 32bits | 32 bits (inferiores) |
| Long | 32bits x 32bits | 64bits. |

[Descripción](http://ciii.frc.utn.edu.ar/TecnicasDigitalesII/WebHome/ASMIntroduccionARM/ASMMultiplicacionARM)

## **Instrucciones de transferencias del registro Status**

Las instrucciones de transferencias del registro Status, transfieren desde o hacia CPSR o SPSR con un registro de propósitos múltiples.

Modificando el CPSR se puede:

* establecer el valor de las banderas de condición.
* establecer el valor de los bits de interrupciones.
* establecer el modo y estado del procesador.

[Descripción](http://ciii.frc.utn.edu.ar/TecnicasDigitalesII/WebHome/ASMIntroduccionARM/ASMInstruccionesStatusARM)

## **Instrucciones de Carga y Almacenamiento**

Las Instrucciones de carga y almacenamiento, realizan transferencias de datos entre los registros de propósitos múltiples y la memoria.

Se dividen en:

* Carga y Almacenamiento de Registros.
* Carga y Almacenamiento de Registros Múltiples.

Existen también instrucciones de intercambio de bytes, pero no se recomienda su uso por estar consideradas obsoletas en versiones posteriores del ARM.

### **Carga y Almacenamiento de Registros**

Una instrucción de carga, permiten cargar un word (32bits), un halfword (16bits), o un byte (8bits) desde memoria a un registro, los bytes o halfword pueden ser automáticamente completados con cero o signo cuando son cargados, dependiendo de la instrucción.

Una instrucción de almacenamiento, permite guardar en memoria un word (32bits), un halfword (16bits), o un byte (8bits) desde un registro.

**Debe tenerse en cuenta que al guardar o cargar un word (32bits) la memoria debe estar alineada a 32 bits, caso contrario, puede afectar al valor guardado o leído de manera impredecible.**

Con respecto al direccionamiento de estas instrucciones, tienen tres modos, todos utilizan un registro base y un corrimiento especificados en la instrucción:

* **Direccionamiento por corrimiento**, la dirección de memoria está formada por la suma o resta del corrimiento con el registro base.
* **Direccionamiento pre-indexado**, la dirección de memoria está formada de la misma forma que la anterior, pero esta dirección de memoria calculada, es escrita luego en el registro base.
* **Direccionamiento post-indexado**, se toma el dato de la dirección apuntada por el registro base y luego se actualiza este sumándole o restándole el corrimiento.

En todos los casos, el corrimiento puede ser un valor inmediato, un registro de propósitos múltiples o un registro de propósitos múltiples con un corrimiento fijo.

Como el PC es un registro de propósito general un valor de 32 bits puede ser cargado directamente de memoria, pudiendo de esta forma acceder a cualquiera de los 4GB del mapa

[Descripción](http://ciii.frc.utn.edu.ar/TecnicasDigitalesII/WebHome/ASMIntroduccionARM/ASMInstruccionesCyAARM)

### **Carga y Almacenamiento de Registros Múltiples**

Las instrucciones de Carga Múltiple (LDM) y Almacenamiento Múltiple (STM), realizan una transferencia de un conjunto de registro de propósitos múltiples hacia o desde la la memoria.

Existen cuatro modos de direccionamiento

* pre-incremento.
* post-incremento.
* pre-decremento.
* post-decremento.

El registro base es especificado por un registro, el cual puede ser opcionalmente actualizando después de la transferencia.

En el caso de la subrutinas, el uso de las LDM y STM en las entrada y salida es utilizado de la siguiente forma:

* Una instrucción STM en la entrada de la subrutina puede guardar los registros en el stack, actualizando la dirección del mismo.
* Una instrucción LDM en la salida de la subrutina puede recuperar el contenido de los registros desde el stack, cargar el PC con la dirección de retorno y actualizar la dirección del stack.

Las instrucciones LDM y STM permite construir eficientes código para copiar y/o procesar bloques de datos.

[Descripción](http://ciii.frc.utn.edu.ar/TecnicasDigitalesII/WebHome/ASMIntroduccionARM/ASMInstruccionesCyAMARM)

## **Instrucciones para la generación de excepciones**

Existen dos tipos de instrucciones designadas para causar una excepción específica

### **Instrucciones de interrupción por software**

Las instrucciones SWI, causan que ocurra una excepción por interrupción de software.

Ellas son usadas normalmente para realizar llamadas al sistema operativo, requiriendo un servicio definido por el SO.

Esta excepción causan también un cambio en los privilegios del modo del procesador, esto permite realizar acceso a tareas con privilegios, pero solo a través del control del SO.

### **Instrucción de Breakpoint por Software**

Las instrucciones BKPT genera una excepción de aborto.

En el caso de tener un debugger instalado en el vector de aborto, una excepción generada de esta forma es tratada como un breakpoint.

Si esta presente en el sistema un debug por hardware, esto puede dar lugar directamente al tratamiento de la BKPT como un breakpoint, previniendo que se produzca efectivamente la excepción de aborto.

Para terminar, los siguientes tipos de instrucciones causan un excepción indefinida:

* Una instrucción de coprocesador la cual no es reconocida por el hardware del coprocesador.
* Word que no tienen asignada una instrucción

Las excepciones indefinidas, son normalmente usadas, ya sea para generar un error o para iniciar el software de emulación de la instrucción.