# Instrucciones Implementadas

El subconjunto de THUMB-2 de instrucciones implementadas será descrito en las siguientes páginas.

Tipos de instrucciones:

* Operaciones con inmediatos
* Operaciones con registro
* Lectura/Escritura
* Salto

## Operaciones con inmediatos:

* Suma/Resta con inmediato de 12 bits sin modificar
* Mover con inmediato de 16 bits sin modificar

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instrucción | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Procesado de datos (Inmediato) | 1 | 1 | 1 | 1 | 0 |  | | | | | | | | | | |
| Suma/Resta, inmediato de 12-bit |  | | | | | i | 1 | 0 | OP | 0 | OP2 | | Rn | | | |
| Mover, inmediato de 16-bit | i | 1 | 0 | OP | 1 | OP2 | | imm4 | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instrucción | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Procesado de datos (Inmediato) | 0 |  | | | | | | | | | | | | | | |
| Suma/ Resta, inmediato de 12-bit |  | imm3 | | | Rd | | | | imm8 | | | | | | | |
| Mover, inmediato de 16-bit | imm3 | | | Rd | | | | imm8 | | | | | | | |

Estas instrucciones se identifican por los bits [31 a 27] y [15], estos deben ser “11110” y “0”. El bit 22 diferencia entre una operación de suma/resta ([22]=0) y una operación mover ([22]=1).

1. Suma, Resta, inmediato de 12-bit:

Suma o resta de un inmediato al valor de un registro. El valor calculado se conserva en otro registro. El inmediato es un número con signo, por ello el valor máximo del inmediato es 2047 y el mínimo es -2048.

Las operaciones permitidas son la suma con código de operación “0b000”, y la resta con código de operación “0b110”.

Campos:

* I:imm3:imm8 : Número inmediato de 12 bits
* Rn: Registro origen del primer operando.
* Rd: Registro destino del resultado.
* OP, OP2: Operación a realizar por el procesador.

1. Mover, inmediato de 16-bit:

Copia un inmediato de 16 bits a la mitad elegida de un registro. Es necesario realizar dos operaciones para garantizar que el valor del registro es el deseado.

Se permite mover un inmediato de 16 bits a la mitad más significativa de la palabra con código de operación “0b100”, o a la mitad menos significativa con código de operación “0b000”.

Campos:

* I:imm3:imm8 : Número inmediato de 16 bits
* Rd: Registro destino de la operación.
* OP, OP2: Operación a realizar por el procesador.

## Operaciones con registros

* Procesado de datos: Desplazamiento constante

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instrucción | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Procesado de datos (No inmediatos) | 1 | 1 | 1 |  | 1 | 0 | 1 |  | | | | | | | | |
| Procesado de datos: Desp. constante |  | | | 0 |  | | | OP | | | | S | Rn | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instrucción | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Procesado de datos (No inmediatos) |  | | | | | | | | | | | | | | | |
| Procesado de datos: Desp. constante | SBZ | imm3 | | | Rd | | | | imm2 | | type | | Rm | | | |

Para identificar las operaciones con registros los bits [31 a 29] y [27 a 25] deben ser “111” y “101” respectivamente.

1. Procesado de datos: Desplazamiento constante

Aplica operaciones aritmético-lógicas sobre los operandos obtenidos de dos registros.

Campos:

* OP: Código de operación.
* S: Habilita la activación de flags en la ALU.
* Rn: Registro A, primer operando de la operación.
* Rd: Registro destino para el resultado de la operación.
* Rm: Registro B, segundo operando de la operación.
* Imm3, imm2, type: no usados en esta implementación.

El campo OP, bits [24:21], indican la operación aritmético-lógica aplicada.

Operaciones:

* ADD: Suma el valor de dos registros.
* AND: Aplica la operación lógica “AND” a dos registros.
* CMP: Compara dos registros y activa los flags de la ALU. El campo “Rd” debe ser “1111”. El campo “S” debe ser “1”.
* EOR: Realiza la OR exclusiva entre dos registros.
* MOV: Copia el valor de un registro a otro. El campo “Rn” debe ser “1111”.
* ORR: Aplica la operación lógica “OR” a dos registros.
* SUB: Calcula la diferencia entre dos registro.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Operación | OP | | | |
| ADD | 1 | 0 | 0 | 0 |
| AND | 0 | 0 | 0 | 0 |
| CMP\* | 1 | 1 | 0 | 1 |
| EOR | 0 | 1 | 0 | 0 |
| MOV\* | 0 | 0 | 1 | 0 |
| ORR | 0 | 0 | 1 | 0 |
| SUB | 1 | 1 | 0 | 1 |

## Lectura/Escritura de memoria

* Procesado de datos: Desplazamiento constante

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instrucción | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Lectura y escritura de un dato | 1 | 1 | 1 | 1 | 1 | 0 | 0 |  | | | | | | | | |
| Procesado de datos: Desp. constante |  | | | | | | | S | 1 | Size | | L | Rn | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instrucción | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Lectura y escritura de un dato |  | | | | | | | | | | | | | | | |
| Procesado de datos: Desp. constante | Rt | | | | imm12 | | | | | | | | | | | |

Para detectar una instrucción de acceso a memoria los bits [31 a 25] deben ser “1111100”.

1. Procesado de datos: Desplazamiento constante

Lee de memoria o escribe en memoria un único dato.

Campos:

* S: Indica si el inmediato debe extenderse con signo (1) o con ceros (0).
* L: Indica si la operación es de escritura (0) o de lectura (1).
* Rn: Dirección de lectura/escritura.
* Rt: Registro con los datos de escritura o registro destino para los datos leídos.
* Imm12: Desplazamiento respecto a la dirección de lectura/escritura.
* Size: Tamaño de datos a escribir o leer. No utilizado en esta implementación.

## Saltos

* Salto incondicional
* Salto condicional

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instrucción | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Saltos | 1 | 1 | 1 | 1 | 0 |  | | | | | | | | | | |
| Salto incondicional |  | | | | | S | offset[21:12] | | | | | | | | | |
| Salto condicional | S | Cond | | | | offset[17:12] | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instrucción | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Saltos | 1 |  | | | | | | | | | | | | | | |
| Salto incondicional |  | 0 | I1 | 1 | I2 | offset[11:1] | | | | | | | | | | |
| Salto condicional | 0 | J1 | 0 | J2 | offset[11:1] | | | | | | | | | | |

Las instrucciones de salto deben tener los bits [31 a 27] y [15] a “11110” y “1” respectivamente. El bit [12] indica si se trata de un salto condicional o de un salto incondicional.

Al tratarse de una arquitectura compatible con instrucciones de 16 bits, el bit [0] del offset del salto siempre será “0”.

1. Salto incondicional

Fuerza al programa a tomar un salto a partir del valor del contador de programa y un offset.

Campos:

* S: Indica el signo del offset, se duplica para rellenar los bits [31 a 24] del mismo.
* Offset: Offset del salto.
* I1, I2: bits [23] y [22] del offset.

1. Salto condicional

Se toma el salto si se cumplen las condiciones de la instrucción.

Campos:

* S: Indica el signo del offset, se duplica para rellenar los bits [31 a 20] del mismo.
* Cond: Condiciones para tomar o no el salto.
* Offset: Offset del salto.
* I1, I2: bits [19] y [18] del offset.

Insertar table de condiciones para salto

Bibliografía:

ARM Architecture Reference Manual, Thumb-2 Supplement

<http://read.pudn.com/downloads159/doc/709030/Thumb-2SupplementReferenceManual.pdf>

<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.qrc0001m/index.html>

<https://ece.uwaterloo.ca/~ece222/ARM/ARM7-TDMI-manual-pt3.pdf>

<http://www.sciencezero.org/index.php?title=ARM:_Cortex-M3_Thumb-2_instruction_set>

<https://ece.uwaterloo.ca/~ece222/ARM/ARM7-TDMI-manual-pt3.pdf>