

Cortex -M0

Generado por Doxygen 1.8.7

Viernes, 18 de Septiembre de 2015 23:31:11

Índice general

1	Cortex-M0	1
2	Índice de estructura de datos	3
2.1	Estructura de datos	3
3	Índice de archivos	5
3.1	Lista de archivos	5
4	Documentación de las estructuras de datos	7
4.1	Referencia de la Estructura ins_t	7
4.1.1	Documentación de los campos	7
4.1.1.1	array	7
4.2	Referencia de la Estructura instruction_t	7
4.2.1	Documentación de los campos	7
4.2.1.1	mnemonic	7
4.2.1.2	op1_type	7
4.2.1.3	op1_value	8
4.2.1.4	op2_type	8
4.2.1.5	op2_value	8
4.2.1.6	op3_type	8
4.2.1.7	op3_value	8
5	Documentación de archivos	9
5.1	Referencia del Archivo branch.c	9
5.1.1	Documentación de las funciones	10
5.1.1.1	BAL	10
5.1.1.2	BCC	10
5.1.1.3	BCS	10
5.1.1.4	BEQ	10
5.1.1.5	BGE	11
5.1.1.6	BGT	12
5.1.1.7	BHI	12
5.1.1.8	BLE	12

5.1.1.9	BLS	12
5.1.1.10	BLT	13
5.1.1.11	BMI	13
5.1.1.12	BNE	13
5.1.1.13	BPL	13
5.1.1.14	BVC	14
5.1.1.15	BVS	14
5.2	Referencia del Archivo branch.h	14
5.2.1	Documentación de las funciones	15
5.2.1.1	BAL	15
5.2.1.2	BCC	15
5.2.1.3	BCS	15
5.2.1.4	BEQ	16
5.2.1.5	BGE	16
5.2.1.6	BGT	16
5.2.1.7	BHI	16
5.2.1.8	BL	17
5.2.1.9	BLE	17
5.2.1.10	BLS	17
5.2.1.11	BLT	17
5.2.1.12	BMI	18
5.2.1.13	BNE	18
5.2.1.14	BPL	18
5.2.1.15	BVC	18
5.2.1.16	BVS	19
5.2.1.17	BX	19
5.3	Referencia del Archivo decoder.c	19
5.3.1	Documentación de las funciones	20
5.3.1.1	countLines	20
5.3.1.2	decodeInstruction	20
5.3.1.3	getInstruction	21
5.3.1.4	readFile	21
5.4	Referencia del Archivo decoder.h	21
5.4.1	Documentación de las funciones	21
5.4.1.1	countLines	21
5.4.1.2	decodeInstruction	21
5.4.1.3	getInstruction	22
5.4.1.4	readFile	22
5.5	Referencia del Archivo instrucciones.c	22
5.5.1	Documentación de las funciones	23

5.5.1.1	ADD	23
5.5.1.2	ADDS	23
5.5.1.3	AND	23
5.5.1.4	ASRS	24
5.5.1.5	Bandera	24
5.5.1.6	BIC	24
5.5.1.7	EOR	24
5.5.1.8	LSL	24
5.5.1.9	LSLS	25
5.5.1.10	LSR	25
5.5.1.11	LSRS	25
5.5.1.12	MOV	25
5.5.1.13	MVN	26
5.5.1.14	NOP	26
5.5.1.15	ORR	26
5.5.1.16	REV	26
5.5.1.17	REV16	26
5.5.1.18	ROR	27
5.5.1.19	RSBS	27
5.5.1.20	SUB	27
5.5.2	Documentación de las variables	27
5.5.2.1	i	27
5.6	Referencia del Archivo instrucciones.h	27
5.6.1	Documentación de las funciones	29
5.6.1.1	ADD	29
5.6.1.2	ADDS	30
5.6.1.3	AND	30
5.6.1.4	ASR	30
5.6.1.5	ASRS	30
5.6.1.6	Bandera	31
5.6.1.7	BICS	31
5.6.1.8	CMN	31
5.6.1.9	CMP	31
5.6.1.10	CMPS	32
5.6.1.11	EOR	32
5.6.1.12	LSL	32
5.6.1.13	LSLS	32
5.6.1.14	LSR	33
5.6.1.15	LSRS	33
5.6.1.16	MOV	33

5.6.1.17	MUL	33
5.6.1.18	MVN	34
5.6.1.19	NOP	34
5.6.1.20	ORR	34
5.6.1.21	REV	34
5.6.1.22	REV16	35
5.6.1.23	REVSH	36
5.6.1.24	ROR	36
5.6.1.25	RSB	36
5.6.1.26	SBC	36
5.6.1.27	SUB	37
5.6.1.28	TST	37
5.7	Referencia del Archivo main.c	37
5.7.1	Documentación de las funciones	37
5.7.1.1	main	37
5.8	Referencia del Archivo README.md	38
5.9	Referencia del Archivo test.c	38
5.10	Referencia del Archivo visualizacion.c	38
5.10.1	Documentación de las funciones	38
5.10.1.1	visualizacion_registro	38
5.11	Referencia del Archivo visualizacion.h	38
5.11.1	Documentación de las funciones	38
5.11.1.1	visualizacion_registro	38

Capítulo 1

Cortex-M0

Emulador Procesador Cortex M0 en C Documentación: Nicolás Arias. Desarrollador: Andrés Felipe Sánchez

Capítulo 2

Índice de estructura de datos

2.1. Estructura de datos

Lista de estructuras con una breve descripción:

ins_t	7
instruction_t	7

Capítulo 3

Indice de archivos

3.1. Lista de archivos

Lista de todos los archivos con descripciones breves:

branch.c	9
branch.h	14
decoder.c	19
decoder.h	21
instrucciones.c	22
instrucciones.h	27
main.c	37
test.c	38
visualizacion.c	38
visualizacion.h	38

Capítulo 4

Documentación de las estructuras de datos

4.1. Referencia de la Estructura ins_t

```
#include <decoder.h>
```

Campos de datos

- char ** [array](#)

4.1.1. Documentación de los campos

4.1.1.1. char** array

La documentación para esta estructura fue generada a partir del siguiente fichero:

- [decoder.h](#)

4.2. Referencia de la Estructura instruction_t

```
#include <decoder.h>
```

Campos de datos

- char [mnemonic](#) [10]
- char [op1_type](#)
- char [op2_type](#)
- char [op3_type](#)
- uint32_t [op1_value](#)
- uint32_t [op2_value](#)
- uint32_t [op3_value](#)

4.2.1. Documentación de los campos

4.2.1.1. char mnemonic[10]

4.2.1.2. char op1_type

4.2.1.3. `uint32_t op1_value`

4.2.1.4. `char op2_type`

4.2.1.5. `uint32_t op2_value`

4.2.1.6. `char op3_type`

4.2.1.7. `uint32_t op3_value`

La documentación para esta estructura fue generada a partir del siguiente fichero:

- [decoder.h](#)

Capítulo 5

Documentación de archivos

5.1. Referencia del Archivo branch.c

```
#include "branch.h"
```

Funciones

- void **BEQ** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es igual.
- void **BNE** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es diferente.
- void **BCS** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es mayor o igual (sin signo)
- void **BCC** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es menor (sin signo)
- void **BMI** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es negativo.
- void **BPL** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es positivo.
- void **BVS** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si hay sobreflujo.
- void **BVC** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si no hay sobreflujo.
- void **BHI** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es mayor (sin signo)
- void **BLS** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es menor o igual (sin signo)
- void **BGE** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que genera un salto si es mayor o igual (con signo)
- void **BLT** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es menor (con signo)
- void **BGT** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es mayor (con signo)
- void **BLE** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es menor o igual (con signo)
- void **BAL** (unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto sin condicion, de cualquier forma.

5.1.1. Documentación de las funciones

5.1.1.1. void BAL (unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto sin condicion, de cualquier forma.

Parámetros

<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void.

5.1.1.2. void BCC (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcin que reliza un salto si es menor (sin signo)

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.1.1.3. void BCS (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que reliza un salto si es mayor o igual (sin signo)

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.1.1.4. void BEQ (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si es igual.

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.1.1.5. void BGE (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que genera un salto si es mayor o igual (con signo)

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.1.1.6. void BGT (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si es mayor (con signo)

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void.

5.1.1.7. void BHI (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si es mayor (sin signo)

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.1.1.8. void BLE (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si es menor o igual (con signo)

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.1.1.9. void BLS (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si es menor o igual (sin signo)

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.1.1.10. void BLT (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si es menor (con signo)

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.1.1.11. void BMI (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si es negativo.

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.1.1.12. void BNE (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que reliza un salto si es diferente.

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.1.1.13. void BPL (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si es positivo.

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.1.1.14. void BVC (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si no hay sobreflujo.

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void.

5.1.1.15. void BVS (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si hay sobreflujo.

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.2. Referencia del Archivo branch.h

Funciones

- void **BEQ** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es igual.
- void **BNE** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que reliza un salto si es diferente.
- void **BCS** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que reliza un salto si es mayor o igual (sin signo)
- void **BCC** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcin que reliza un salto si es menor (sin signo)
- void **BMI** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es negativo.
- void **BPL** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es positivo.

- void **BVS** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si hay sobreflujo.
- void **BVC** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si no hay sobreflujo.
- void **BHI** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es mayor (sin signo)
- void **BLS** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es menor o igual (sin signo)
- void **BGE** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que genera un salto si es mayor o igual (con signo)
- void **BLT** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es menor (con signo)
- void **BGT** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es mayor (con signo)
- void **BLE** (unsigned long *Banderas, unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto si es menor o igual (con signo)
- void **BAL** (unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto sin condicion, de cualquier forma.
- void **BL** (unsigned long *PC, unsigned long Posicion)
Funcion que realiza un salto guardando posicion instruccion siguiente en LR.
- void **BX** (unsigned long *PC)
Funcion que realiza un salto a la posicion guardada en LR.

5.2.1. Documentación de las funciones

5.2.1.1. void BAL (unsigned long * PC, unsigned long Posicion)

Funcion que realiza un salto sin condicion, de cualquier forma.

Parámetros

<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void.

5.2.1.2. void BCC (unsigned long * Banderas, unsigned long * PC, unsigned long Posicion)

Funcion que realiza un salto si es menor (sin signo)

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.2.1.3. void BCS (unsigned long * Banderas, unsigned long * PC, unsigned long Posicion)

Funcion que realiza un salto si es mayor o igual (sin signo)

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.2.1.4. void BEQ (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si es igual.

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.2.1.5. void BGE (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que genera un salto si es mayor o igual (con signo)

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.2.1.6. void BGT (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si es mayor (con signo)

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void.

5.2.1.7. void BHI (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si es mayor (sin signo)

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.2.1.8. void BL (unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto guardando posicion instruccion siguiente en LR.

Parámetros

<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.2.1.9. void BLE (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si es menor o igual (con signo)

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.2.1.10. void BLS (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si es menor o igual (sin signo)

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.2.1.11. void BLT (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si es menor (con signo)

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.2.1.12. void BMI (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si es negativo.

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.2.1.13. void BNE (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que reliza un salto si es diferente.

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.2.1.14. void BPL (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si es positivo.

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.2.1.15. void BVC (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si no hay sobreflujo.

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void.

5.2.1.16. void BVS (unsigned long * *Banderas*, unsigned long * *PC*, unsigned long *Posicion*)

Funcion que realiza un salto si hay sobreflujo.

Parámetros

<i>Banderas</i>	Puntero para el registro de las banderas
<i>PC</i>	Puntero a el registro PC
<i>Posicion</i>	Siguiente posicion

Devuelve

void

5.2.1.17. void BX (unsigned long * *PC*)

Funcion que realiza un salto a la posicion guardada en LR.

Parámetros

<i>*PC</i>	Puntero a el registro PC
------------	--------------------------

Devuelve

void

5.3. Referencia del Archivo decoder.c

```
#include "decoder.h"
```

Funciones

- void `decodeInstruction` (`instruction_t` instruction)
Decodifica la instrucción y la ejecuta.
- `instruction_t` `getInstruction` (char *instStr)
Obtiene la instrucción separada por partes.
- int `readFile` (char *filename, `ins_t` *instructions)
- int `countLines` (FILE *fp)

5.3.1. Documentación de las funciones

5.3.1.1. `int countLines (FILE * fp)`

5.3.1.2. `void decodeInstruction (instruction_t instruction)`

Decodifica la instrucción y la ejecuta.

Parámetros

<i>instruction</i>	instrucción a decodificar y ejecutar.
--------------------	---------------------------------------

5.3.1.3. `instruction_t getInstruction (char * instStr)`

Obtiene la instrucción separada por partes.

Parámetros

<i>instStr</i>	cadena que contiene la instrucción.
----------------	-------------------------------------

Devuelve

`instruction_t` la instrucción separada por partes.

5.3.1.4. `int readFile (char * filename, ins_t * instructions)`

5.4. Referencia del Archivo decoder.h

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <stdint.h>
```

Estructuras de datos

- struct `ins_t`
- struct `instruction_t`

Funciones

- void `decodeInstruction (instruction_t instruction)`
Decodifica la instrucción y la ejecuta.
- `instruction_t getInstruction (char *instStr)`
Obtiene la instrucción separada por partes.
- int `readFile (char *filename, ins_t *instructions)`
- int `countLines (FILE *fp)`

5.4.1. Documentación de las funciones

5.4.1.1. `int countLines (FILE * fp)`5.4.1.2. `void decodeInstruction (instruction_t instruction)`

Decodifica la instrucción y la ejecuta.

Parámetros

<i>instruction</i>	instrucción a decodificar y ejecutar.
--------------------	---------------------------------------

5.4.1.3. `instruction_t getInstruction (char * instStr)`

Obtiene la instrucción separada por partes.

Parámetros

<i>instrStr</i>	cadena que contiene la instrucción.
-----------------	-------------------------------------

Devuelve

`instruction_t` la instrucción separada por partes.

5.4.1.4. `int readFile (char * filename, ins_t * instructions)`

5.5. Referencia del Archivo instrucciones.c

```
#include "instrucciones.h"
```

Funciones

- void `ADD` (unsigned long *Rd, unsigned long *Rn, unsigned long Num)
Funcion para realizar la suma de un registro y un valor.
- void `ADDS` (unsigned long *Rd, unsigned long *Rn, unsigned long *Rm)
Funcion para realizar la suma con dos registros.
- void `ORR` (unsigned long *Rd, unsigned long *Rn, unsigned long *Rm)
Funcion para realizar ORR con dos registros.
- void `EOR` (unsigned long *Rd, unsigned long *Rn, unsigned long *Rm)
Funcion para realizar EOR con dos registros.
- void `MOV` (unsigned long *Rd, unsigned long *Rn)
Funcion para realizar MOV (reemplazar el valor de un registro en otro)
- void `AND` (unsigned long *Rd, unsigned long *Rn, unsigned long *Rm)
Funcion para realizar AND dos registros.
- void `SUB` (unsigned long *Rd, unsigned long *Rn, unsigned long *Rm)
Funcion para realizar la resta entre un registro y un valor.
- void `Bandera` (unsigned long *Rd, unsigned long *Rn, unsigned long *Rm, unsigned long *Banderas)
Funcion generadoras de banderas.
- void `LSL` (unsigned long *Rd, unsigned long *Rn, unsigned long Num)
Funcion desplazamiento hacia la izquierda ingresando un numero a sumar.
- void `LSLS` (unsigned long *Rd, unsigned long *Rn)
- void `LSR` (unsigned long *Rd, unsigned long *Rn, unsigned long Num)
Funcion desplazamiento hacia la derecha con un numero a sumar.
- void `LSRS` (unsigned long *Rd, unsigned long *Rn)
Funcion desplazamiento hacia la Derecha con dos registros.
- void `ROR` (unsigned long *Rd, unsigned long *Rn)
Funcion para rotar.
- void `ASRS` (unsigned long *Rd, unsigned long *Rn)

Funcion para despalazar aritmeticamente hacia la derecha con dos registros.

- void **REV** (unsigned long *Rd, unsigned long *Rn)

Funcion para cambiar el orden de los bits.

- void **REV16** (unsigned long *Rd, unsigned long *Rn)

Funcion para modificar la posicion de grupos de 16 bits.

- void **BIC** (unsigned long *Rd, unsigned long *Rn)
- void **MVN** (unsigned long *Rd, unsigned long *Rn)

funcion para hacer el complemento a un registro.

- void **RSBS** (unsigned long *Rd, unsigned long *Rn)
- void **NOP** (void)

Funcion que hace un retardo.

Variables

- int **i**

5.5.1. Documentación de las funciones

5.5.1.1. void ADD (unsigned long * Rd, unsigned long * Rn, unsigned long Num)

Funcion para realizar la suma de un registro y un valor.

Parámetros

<i>Rd</i>	Registro para el resultado.
<i>Rn</i>	Registro de ingreso para sumar.
<i>Num</i>	Valor a sumar.

Devuelve

void

5.5.1.2. void ADDS (unsigned long * Rd, unsigned long * Rn, unsigned long * Rm)

Funcion para realizar la suma con dos registros.

Parámetros

<i>Rd</i>	Registro para el resultado.
<i>Rn</i>	Primer registro de ingreso a sumar.
<i>Rm</i>	Segundo registro de ingreso a sumar.

Devuelve

void

5.5.1.3. void AND (unsigned long * Rd, unsigned long * Rn, unsigned long * Rm)

Funcion para realizar AND dos registros.

Parámetros

<i>Rd</i>	Registro para el resultado
<i>Rn</i>	Primer registro de ingreso para AND
<i>Rm</i>	Segundo registro de ingreso para AND

Devuelve

void

5.5.1.4. void ASRS (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion para despalazar aritmeticamente hacia la derecha con dos registros.

Parámetros

<i>Rd</i>	Resultado de la operacion.
<i>Rn</i>	numero a trasladar.

Devuelve

void.

5.5.1.5. void Bandera (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long * *Rm*, unsigned long * *Banderas*)

Funcion generadoras de banderas.

Parámetros

<i>Rd</i>	Resultado de la operacion.
<i>Rn</i>	Primer valor de operacion.
<i>Rm</i>	Segundo valor de operacion.
<i>Banderas</i>	Es un arreglo donde se guardaran los resutados de da las banderas.

Devuelve

void

5.5.1.6. void BIC (unsigned long * *Rd*, unsigned long * *Rn*)**5.5.1.7. void EOR (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long * *Rm*)**

Funcion para realizar EOR con dos registros.

Parámetros

<i>Rd</i>	Registro para el resultado.
<i>Rn</i>	Primer registro de ingreso para EOR
<i>Rm</i>	Segundo registro de ingreso para EOR.

Devuelve

void

5.5.1.8. void LSL (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long *Num*)

Funcion desplazamiento hacia la izquierda ingresando un numero a sumar.

Parámetros

<i>Rd</i>	Resultado de la operacion.
<i>Rn</i>	Valor del registro a sumar.
<i>Num</i>	Valor a sumar.

Devuelve

void.

5.5.1.9. void LSLS (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion desplazamiento hacia la izquierda con dos registros.

Parámetros

<i>Rd</i>	Resultado de la operacion.
<i>Rn</i>	Valor del registro a sumar.

Devuelve

void.

5.5.1.10. void LSR (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long *Num*)

Funcion desplazamiento hacia la derecha con un numero a sumar.

Parámetros

<i>Rd</i>	Resultado de la operacion
<i>Rn</i>	Valor del registro a sumar.
<i>Num</i>	Valor a sumar.

Devuelve

void

5.5.1.11. void LSRS (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion desplazamiento hacia la Derecha con dos registros.

Parámetros

<i>Rd</i>	Resultado de la operacion.
<i>Rn</i>	Valor del registro a sumar.

Devuelve

void.

5.5.1.12. void MOV (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion para realizar MOV (reemplazar el valor de un registro en otro)

Parámetros

<i>Rd</i>	Registro para el resultado
<i>Rn</i>	Registro de ingreso para MOV (posicion en la cual se va a mover el valor del registro)

Devuelve

void

5.5.1.13. void MVN (unsigned long * *Rd*, unsigned long * *Rn*)

funcion para hacer el complemento a un registro.

Parámetros

<i>Rd</i>	Resultado de la operacion.
-----------	----------------------------

Devuelve

void.

5.5.1.14. void NOP (void)

Funcion que hace un retardo.

Devuelve

void.

5.5.1.15. void ORR (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long * *Rm*)

Funcion para realizar ORR con dos registros.

Parámetros

<i>Rd</i>	Registro para el resultado
<i>Rn</i>	Primer registro de ingreso para ORR.
<i>Rm</i>	Segundo registro de ingreso para ORR.

Devuelve

void

5.5.1.16. void REV (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion para cambiar el orden de los bits.

Parámetros

<i>Rb</i>	Registro a modificar.
-----------	-----------------------

Devuelve

void.

5.5.1.17. void REV16 (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion para modificar la posicion de grupos de 16 bits.

Parámetros

<i>Rd</i>	Registro resultante.
<i>Rn</i>	Registro a modificar.

Devuelve

void.

5.5.1.18. void ROR (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion para rotar.

Parámetros

<i>Rdn</i>	Resultado de la operacion.
<i>Rn</i>	Numero a rotar.

Devuelve

void

5.5.1.19. void RSBS (unsigned long * *Rd*, unsigned long * *Rn*)5.5.1.20. void SUB (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long * *Rm*)

Funcion para realizar la resta entre un registro y un valor.

Parámetros

<i>Rd</i>	Registro para el resultado
<i>Rn</i>	Primer registro de ingreso a sumar
<i>Rm</i>	Segundo registro de ingreso a sumar

Devuelve

void

5.5.2. Documentación de las variables

5.5.2.1. int i

5.6. Referencia del Archivo instrucciones.h

Funciones

- void **ADD** (unsigned long **Rd*, unsigned long **Rn*, unsigned long Num)
Funcion para realizar la suma de un registro y un valor.
- void **ADDS** (unsigned long **Rd*, unsigned long **Rn*, unsigned long **Rm*)
Funcion para realizar la suma con dos registros.
- void **ORR** (unsigned long **Rd*, unsigned long **Rn*, unsigned long **Rm*)
Funcion para realizar ORR con dos registros.
- void **EOR** (unsigned long **Rd*, unsigned long **Rn*, unsigned long **Rm*)
Funcion para realizar EOR con dos registros.

- void **MOV** (unsigned long *Rd, unsigned long *Rn)
Funcion para realizar MOV (reemplazar el valor de un registro en otro)
- void **AND** (unsigned long *Rd, unsigned long *Rn, unsigned long *Rm)
Funcion para realizar AND dos registros.
- void **SUB** (unsigned long *Rd, unsigned long *Rn, unsigned long *Rm)
Funcion para realizar la resta entre un registro y un valor.
- void **Bandera** (unsigned long *Rd, unsigned long *Rn, unsigned long *Rm, unsigned long *Banderas)
Funcion generadoras de banderas.
- void **LSL** (unsigned long *Rd, unsigned long *Rn, unsigned long Num)
Funcion desplazamiento hacia la izquierda ingresando un numero a sumar.
- void **LSLS** (unsigned long *Rd, unsigned long *Rn)
- void **LSR** (unsigned long *Rd, unsigned long *Rn, unsigned long Num)
Funcion desplazamiento hacia la derecha con un numero a sumar.
- void **LSRS** (unsigned long *Rd, unsigned long *Rn)
Funcion desplazamiento hacia la Derecha con dos registros.
- void **ROR** (unsigned long *Rd, unsigned long *Rn)
Funcion para rotar.
- void **ASR** (unsigned long *Rd, unsigned long *Rn, unsigned long Num)
Funcion para desplazar aritmeticamente hacia la derecha con un registro y y un numero.
- void **ASRS** (unsigned long *Rd, unsigned long *Rn)
Funcion para despalazar aritmeticamente hacia la derecha con dos registros.
- void **BICS** (unsigned long *Rd, unsigned long *Rn)
Funcion que hace una and de 1 bit y con Rm negado.
- void **CMN** (unsigned long *Rn, unsigned long *Rm)
Funcion que modifica las banderas sin guardar el resultado.
- void **CMP** (unsigned long *Rn, unsigned long *Rm)
Funcion que modifica las banderas comparando los registros.
- void **CMPS** (unsigned long *Rn, unsigned long Num)
Funcion que modifica banderas comparando un registro y un numero.
- void **MUL** (unsigned long *Rd, unsigned long Rn, unsigned long Rm)
Funcion Multiplica dos registros, sin signo y almacena los 32 bits menos significativos del resultado en el regsitro de destino.
- void **MVN** (unsigned long *Rd, unsigned long *Rn)
funcion para hacer el complemento a un registro.
- void **NOP** (void)
Funcion que hace un retardo.
- void **REV** (unsigned long *Rd, unsigned long *Rn)
Funcion para cambiar el orden de los bits.
- void **REV16** (unsigned long *Rd, unsigned long *Rn)
Funcion para modificar la posicion de grupos de 16 bits.
- void **REVSH** (unsigned long *Rd, unsigned long *Rn)
Funcion para modificar la posicion de a ocho bits.
- void **RSB** (unsigned long *Rd)
Funcion para obtener el complemento a dos de un numero.
- void **SBC** (unsigned long *Rd, unsigned long *Rn, unsigned long *Rm)
Funcion para restar con Carry.
- void **TST** (unsigned long *Rd, unsigned long *Rn)
Funcion que hace una AND bit a bit, pero solo modifica las banderas.

5.6.1. Documentación de las funciones

5.6.1.1. void ADD (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long *Num*)

Funcion para realizar la suma de un registro y un valor.

Parámetros

<i>Rd</i>	Registro para el resultado.
<i>Rn</i>	Registro de ingreso para sumar.
<i>Num</i>	Valor a sumar.

Devuelve

void

5.6.1.2. void ADDS (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long * *Rm*)

Funcion para realizar la suma con dos registros.

Parámetros

<i>Rd</i>	Registro para el resultado.
<i>Rn</i>	Primer registro de ingreso a sumar.
<i>Rm</i>	Segundo registro de ingreso a sumar.

Devuelve

void

5.6.1.3. void AND (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long * *Rm*)

Funcion para realizar AND dos registros.

Parámetros

<i>Rd</i>	Registro para el resultado
<i>Rn</i>	Primer registro de ingreso para AND
<i>Rm</i>	Segundo registro de ingreso para AND

Devuelve

void

5.6.1.4. void ASR (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long *Num*)

Funcion para desplazar aritmeticamente hacia la derecha con un registro y y un numero.

Parámetros

<i>Rd</i>	Resultado de la operacion.
<i>Rn</i>	Valor del registro a sumar.
<i>Num</i>	Valor a sumar.

Devuelve

void.

5.6.1.5. void ASRS (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion para despalazar aritmeticamente hacia la derecha con dos registros.

Parámetros

<i>Rd</i>	Resultado de la operacion.
<i>Rn</i>	numero a trasladar.

Devuelve

void.

5.6.1.6. void Bandera (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long * *Rm*, unsigned long * *Banderas*)

Funcion generadoras de banderas.

Parámetros

<i>Rd</i>	Resultado de la operacion.
<i>Rn</i>	Primer valor de operacion.
<i>Rm</i>	Segundo valor de operacion.
<i>Banderas</i>	Es un arreglo donde se guardaran los resutados de da las banderas.

Devuelve

void

5.6.1.7. void BICS (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion que hace una and de 1 bit y con Rm negado.

Parámetros

<i>Rd</i>	Resultado de la operacion
<i>Rn</i>	Registro a negar.

Devuelve

void.

5.6.1.8. void CMN (unsigned long * *Rn*, unsigned long * *Rm*)

Funcion que modifica las banderas sin guardar el resultado.

Parámetros

<i>Rn</i>	Primer registro a operar.
<i>Rm</i>	Segundo registro a operar.

Devuelve

void.

5.6.1.9. void CMP (unsigned long * *Rn*, unsigned long * *Rm*)

Funcion que modifica las banderas comparando los registros.

Parámetros

<i>Rn</i>	Primer registro a comparar.
<i>Rm</i>	Segundo registro a comparar.

Devuelve

void.

5.6.1.10. void CMPS (unsigned long * *Rn*, unsigned long *Num*)

Funcion que modifica banderas comparando un registro y un numero.

Parámetros

<i>Rn</i>	Registro a comparar.
<i>Num</i>	Numero para comparar con el registro.

Devuelve

void.

5.6.1.11. void EOR (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long * *Rm*)

Funcion para realizar EOR con dos registros.

Parámetros

<i>Rd</i>	Registro para el resultado.
<i>Rn</i>	Primer registro de ingreso para EOR
<i>Rm</i>	Segundo registro de ingreso para EOR.

Devuelve

void

5.6.1.12. void LSL (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long *Num*)

Funcion desplazamiento hacia la izquierda ingresando un numero a sumar.

Parámetros

<i>Rd</i>	Resultado de la operacion.
<i>Rn</i>	Valor del registro a sumar.
<i>Num</i>	Valor a sumar.

Devuelve

void.

5.6.1.13. void LSLS (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion desplazamiento hacia la izquierda con dos registros.

Parámetros

<i>Rd</i>	Resultado de la operacion.
<i>Rn</i>	Valor del registro a sumar.

Devuelve

void.

5.6.1.14. void LSR (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long *Num*)

Funcion desplazamiento hacia la derecha con un numero a sumar.

Parámetros

<i>Rd</i>	Resultado de la operacion
<i>Rn</i>	Valor del registro a sumar.
<i>Num</i>	Valor a sumar.

Devuelve

void

5.6.1.15. void LSRS (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion desplazamiento hacia la Derecha con dos registros.

Parámetros

<i>Rd</i>	Resultado de la operacion.
<i>Rn</i>	Valor del registro a sumar.

Devuelve

void.

5.6.1.16. void MOV (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion para realizar MOV (reemplazar el valor de un registro en otro)

Parámetros

<i>Rd</i>	Registro para el resultado
<i>Rn</i>	Registro de ingreso para MOV (posicion en la cual se va a mover el valor del registro)

Devuelve

void

5.6.1.17. void MUL (unsigned long * *Rd*, unsigned long *Rn*, unsigned long *Rm*)

Funcion Multiplica dos registros, sin signo y almacena los 32 bits menos significativos del resultado en el registro de destino.

Parámetros

<i>Rd</i>	Registro para el resultado de la operacion.
<i>Rn</i>	Primer valor del reistro a multiplicar.
<i>Rm</i>	Segundo valor del registro a multiplicar.

Devuelve

void.

5.6.1.18. void MVN (unsigned long * *Rd*, unsigned long * *Rn*)

funcion para hacer el complemento a un registro.

Parámetros

<i>Rd</i>	Resultado de la operacion.
-----------	----------------------------

Devuelve

void.

5.6.1.19. void NOP (void)

Funcion que hace un retardo.

Devuelve

void.

5.6.1.20. void ORR (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long * *Rm*)

Funcion para realizar ORR con dos registros.

Parámetros

<i>Rd</i>	Registro para el resultado
<i>Rn</i>	Primer registro de ingreso para ORR.
<i>Rm</i>	Segundo registro de ingreso para ORR.

Devuelve

void

5.6.1.21. void REV (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion para cambiar el orden de los bits.

Parámetros

<i>Rb</i>	Registro a modificar.
-----------	-----------------------

Devuelve

void.

5.6.1.22. void REV16 (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion para modificar la posicion de grupos de 16 bits.

Parámetros

<i>Rd</i>	Registro resultante.
<i>Rn</i>	Registro a modificar.

Devuelve

void.

5.6.1.23. void REVSH (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion para modificar la posicion de a ocho bits.

Parámetros

<i>Rd</i>	Numero a modificar.
-----------	---------------------

Devuelve

void

5.6.1.24. void ROR (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion para rotar.

Parámetros

<i>Rdn</i>	Resultado de la operacion.
<i>Rn</i>	Numero a rotar.

Devuelve

void

5.6.1.25. void RSB (unsigned long * *Rd*)

Funcion para obtener el complemento a dos de un numero.

Parámetros

<i>Rb</i>	Registro al cual se le hara el complemento.
-----------	---

Devuelve

void.

5.6.1.26. void SBC (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long * *Rm*)

Funcion para restar con Carry.

Parámetros

<i>Rd</i>	Registro para el resultado de la operacion.
<i>Rn</i>	Primer valor a operar
<i>Rm</i>	Segundo valor a operar

Devuelve

void.

5.6.1.27. void SUB (unsigned long * *Rd*, unsigned long * *Rn*, unsigned long * *Rm*)

Funcion para realizar la resta entre un registro y un valor.

Parámetros

<i>Rd</i>	Registro para el resultado
<i>Rn</i>	Primer registro de ingreso a sumar
<i>Rm</i>	Segundo registro de ingreso a sumar

Devuelve

void

5.6.1.28. void TST (unsigned long * *Rd*, unsigned long * *Rn*)

Funcion que hace una AND bit a bit, pero solo modifica las banderas.

Parámetros

<i>Rn</i>	Primer valor ingresado para hacer la operacion.
<i>Rm</i>	Segundo valor ingresado para hacer la operacion.

Devuelve

void.

5.7. Referencia del Archivo main.c

```
#include <stdio.h>
#include "visualizacion.h"
#include "instrucciones.h"
#include "decoder.h"
```

Funciones

- int `main` ()

5.7.1. Documentación de las funciones

5.7.1.1. int `main` ()

\ Arreglo registro[] Datos recibidos del microcontrolador. \ Variable i Contador.

5.8. Referencia del Archivo README.md

5.9. Referencia del Archivo test.c

```
#include <stdio.h>
#include "visualizacion.h"
#include "instrucciones.h"
```

5.10. Referencia del Archivo visualizacion.c

```
#include <stdio.h>
#include <ncurses.h>
#include "visualizacion.h"
```

Funciones

- void [visualizacion_registro](#) (unsigned long *R, unsigned long *Banderas)

5.10.1. Documentación de las funciones

5.10.1.1. void [visualizacion_registro](#) (unsigned long * *R*, unsigned long * *Banderas*)

\brief Funcion para visualizar los registros \param *R Puntero que señala la posicion de los registros almacenados.
\param *Banderas Corresponde a las banderas Cero, Negativo, Acarreo y Sobreflujo. \return void

5.11. Referencia del Archivo visualizacion.h

Funciones

- void [visualizacion_registro](#) (unsigned long *R, unsigned long *Banderas)

5.11.1. Documentación de las funciones

5.11.1.1. void [visualizacion_registro](#) (unsigned long * *R*, unsigned long * *Banderas*)

\brief Funcion para visualizar los registros \param *R Puntero que señala la posicion de los registros almacenados.
\param *Banderas Corresponde a las banderas Cero, Negativo, Acarreo y Sobreflujo. \return void