

Emulador CortexM0

Generado por Doxygen 1.8.10

Sábado, 26 de Septiembre de 2015 01:51:09

Índice general

1	emulador_arm	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	B(unsigned long **PC, unsigned long numeroSaltos)	10
5.1.1.2	BCC(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	10
5.1.1.3	BCS(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	10
5.1.1.4	BEQ(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	10
5.1.1.5	BGE(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	11
5.1.1.6	BGT(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	11
5.1.1.7	BHI(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	11
5.1.1.8	BL(unsigned long **PC, unsigned long numeroSaltos, unsigned long **LR)	11

5.1.1.9	BLE(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	12
5.1.1.10	BLS(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	12
5.1.1.11	BLT(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	12
5.1.1.12	BLX(unsigned long **PC, unsigned long posicionSiguiente, unsigned long **LR)	12
5.1.1.13	BMI(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	13
5.1.1.14	BNE(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	13
5.1.1.15	BPL(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	13
5.1.1.16	BVC(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	13
5.1.1.17	BVS(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	14
5.1.1.18	BX(unsigned long **PC, unsigned long posicionSiguiente)	14
5.2	Referencia del Archivo branch.h	14
5.2.1	Documentación de las funciones	15
5.2.1.1	B(unsigned long **PC, unsigned long numeroSaltos)	15
5.2.1.2	BCC(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	15
5.2.1.3	BCS(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	16
5.2.1.4	BEQ(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	17
5.2.1.5	BGE(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	17
5.2.1.6	BGT(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	17
5.2.1.7	BHI(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	17
5.2.1.8	BL(unsigned long **PC, unsigned long numeroSaltos, unsigned long **LR)	18
5.2.1.9	BLE(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	18
5.2.1.10	BLS(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	18
5.2.1.11	BLT(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	18
5.2.1.12	BLX(unsigned long **PC, unsigned long posicionSiguiente, unsigned long **LR)	19
5.2.1.13	BMI(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	19
5.2.1.14	BNE(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	19
5.2.1.15	BPL(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	19
5.2.1.16	BVC(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	20
5.2.1.17	BVS(unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)	20
5.2.1.18	BX(unsigned long **PC, unsigned long posicionSiguiente)	20
5.3	Referencia del Archivo decoder.c	21
5.3.1	Documentación de las funciones	21
5.3.1.1	countLines(FILE *fp)	21
5.3.1.2	decodeInstruction(instruction_t instruction, unsigned long *r[], unsigned long *bandera, unsigned long *PC, unsigned long *LR)	21
5.3.1.3	getInstruction(char *instStr)	21
5.3.1.4	readFile(char *filename, ins_t *instructions)	21
5.4	Referencia del Archivo decoder.h	21
5.4.1	Documentación de las funciones	22
5.4.1.1	countLines(FILE *fp)	22

5.4.1.2	decodeInstruction(instruction_t instruction, unsigned long *r[], unsigned long *bandera, unsigned long *PC, unsigned long *LR)	22
5.4.1.3	getInstruction(char *instStr)	22
5.4.1.4	readFile(char *filename, ins_t *instructions)	22
5.5	Referencia del Archivo funciones_pila.c	22
5.5.1	Documentación de las funciones	22
5.5.1.1	POP(unsigned long registros[16], unsigned long *R, unsigned long *memoria)	22
5.5.1.2	PUSH(unsigned long registros[16], unsigned long *R, unsigned long *memoria)	22
5.6	Referencia del Archivo funciones_pila.h	23
5.6.1	Documentación de las funciones	23
5.6.1.1	POP(unsigned long registros[16], unsigned long *R, unsigned long *memoria)	23
5.6.1.2	PUSH(unsigned long registros[16], unsigned long *R, unsigned long *memoria)	23
5.7	Referencia del Archivo instrucciones_desplazamiento.c	23
5.7.1	Documentación de las funciones	24
5.7.1.1	ASR(unsigned long Variable_entrada, unsigned long constante_parametro)	24
5.7.1.2	BIC(unsigned long Variable_entrada)	24
5.7.1.3	LSL(unsigned long Variable_entrada, unsigned long constante_parametro)	24
5.7.1.4	LSR(unsigned long Variable_entrada, unsigned long constante_parametro)	25
5.7.1.5	MVN(unsigned long Variable_entrada)	25
5.7.1.6	NOP()	25
5.7.1.7	REV(unsigned long Variable_entrada)	25
5.7.1.8	REV16(unsigned long Variable_entrada)	25
5.7.1.9	REVSH(unsigned long Variable_entrada)	26
5.7.1.10	ROR(unsigned long Variable_entrada, unsigned long constante_parametro)	26
5.7.1.11	RSBS(unsigned long Variable_entrada, unsigned long constante_parametro)	26
5.8	Referencia del Archivo instrucciones_desplazamiento.h	26
5.8.1	Documentación de las funciones	27
5.8.1.1	ASR(unsigned long Variable_entrada, unsigned long constante_parametro)	27
5.8.1.2	BIC(unsigned long Variable_entrada)	27
5.8.1.3	LSL(unsigned long Variable_entrada, unsigned long constante_parametro)	27
5.8.1.4	LSR(unsigned long Variable_entrada, unsigned long constante_parametro)	28
5.8.1.5	MVN(unsigned long Variable_entrada)	28
5.8.1.6	NOP()	28
5.8.1.7	REV(unsigned long Variable_entrada)	28
5.8.1.8	REV16(unsigned long Variable_entrada)	28
5.8.1.9	REVSH(unsigned long variable_entrada)	29
5.8.1.10	ROR(unsigned long Variable_entrada, unsigned long constante_parametro)	29
5.8.1.11	RSBS(unsigned long Variable_entrada, unsigned long constante_parametro)	29
5.9	Referencia del Archivo mostrar_resultado.c	29
5.9.1	Documentación de las funciones	30

5.9.1.1	mostrar(unsigned long resultado)	30
5.10	Referencia del Archivo mostrar_resultado.h	30
5.10.1	Documentación de las funciones	30
5.10.1.1	mostrar(unsigned long resultado)	30
5.11	Referencia del Archivo operaciones_alu.c	30
5.11.1	Documentación de las funciones	31
5.11.1.1	ADC(unsigned long op1, unsigned long op2, unsigned long **bandera)	31
5.11.1.2	ADD(unsigned long op1, unsigned long op2, unsigned long **bandera)	31
5.11.1.3	AND(unsigned long op1, unsigned long op2, unsigned long **bandera)	31
5.11.1.4	BDES(unsigned long op1, unsigned long **bandera)	31
5.11.1.5	EOR(unsigned long op1, unsigned long op2, unsigned long **bandera)	32
5.11.1.6	MOV(unsigned long op1, unsigned long op2, unsigned long **bandera)	32
5.11.1.7	ORR(unsigned long op1, unsigned long op2, unsigned long **bandera)	32
5.11.1.8	SUB(unsigned long op1, unsigned long op2, unsigned long **bandera)	32
5.11.2	Documentación de las variables	33
5.11.2.1	auxban	33
5.11.2.2	constante	33
5.11.2.3	resultado	33
5.12	Referencia del Archivo operaciones_alu.h	33
5.12.1	Documentación de las funciones	33
5.12.1.1	ADC(unsigned long op1, unsigned long op2, unsigned long **bandera)	33
5.12.1.2	ADD(unsigned long op1, unsigned long op2, unsigned long **bandera)	34
5.12.1.3	AND(unsigned long op1, unsigned long op2, unsigned long **bandera)	34
5.12.1.4	BDES(unsigned long op1, unsigned long **bandera)	34
5.12.1.5	EOR(unsigned long op1, unsigned long op2, unsigned long **bandera)	34
5.12.1.6	MOV(unsigned long op1, unsigned long op2, unsigned long **bandera)	34
5.12.1.7	ORR(unsigned long op1, unsigned long op2, unsigned long **bandera)	35
5.12.1.8	SUB(unsigned long op1, unsigned long op2, unsigned long **bandera)	35
5.13	Referencia del Archivo README.md	35
5.14	Referencia del Archivo registro.c	35
5.14.1	Documentación de las funciones	36
5.14.1.1	mostrar_registros(unsigned long r[13])	36
5.15	Referencia del Archivo registro.h	37
5.15.1	Documentación de las funciones	37
5.15.1.1	mostrar_registros(unsigned long r[13])	37
5.16	Referencia del Archivo test.c	37
5.16.1	Documentación de las funciones	37
5.16.1.1	main()	37

Capítulo 1

emulador_arm

emulacion del procesador cortex M0 echo por : andres ramriez villarreal, jhon aldemar duran, juan david ospina

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	21
decoder.h	21
funciones_pila.c	22
funciones_pila.h	23
instrucciones_desplazamiento.c	23
instrucciones_desplazamiento.h	26
mostrar_resultado.c	29
mostrar_resultado.h	30
operaciones_alu.c	30
operaciones_alu.h	33
registro.c	35
registro.h	37
test.c	37

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 **B** (unsigned long **PC, unsigned long numeroSaltos)
Función encargada de saltar sin condición.
- void **BEQ** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si los datos son iguales.
- void **BNE** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si los datos no son iguales.
- void **BCS** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si un dato es mayor o igual al otro (unsigned)
- void **BCC** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si un dato es menor al otro (unsigned)
- void **BMI** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si el dato es negativo.
- void **BPL** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si el dato es positivo.
- void **BVS** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si hay sobreflujo.
- void **BVC** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si no hay sobreflujo.
- void **BHI** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si un dato es mayor al otro (unsigned)
- void **BLS** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si un dato es menor o igual al otro (unsigned)
- void **BGE** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si un dato es mayor o igual al otro.
- void **BLT** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si un dato es menor al otro.
- void **BGT** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si un dato es mayor al otro.
- void **BLE** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)

Función encargada de saltar si un dato es menor o igual al otro.

- void **BL** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **LR)

Función encargada de saltar sin condición y modificar LR.

- void **BX** (unsigned long **PC, unsigned long posicionSiguiente)

Función encargada de ir a una posición específica sin condición.

- void **BLX** (unsigned long **PC, unsigned long posicionSiguiente, unsigned long **LR)

Función encargada de ir a una posición específica sin condición y modificar LR.

5.1.1. Documentación de las funciones

5.1.1.1. void B (unsigned long ** PC, unsigned long numeroSaltos)

Función encargada de saltar sin condición.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.2. void BCC (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si un dato es menor al otro (unsigned)

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.3. void BCS (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si un dato es mayor o igual al otro (unsigned)

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.4. void BEQ (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si los datos son iguales.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.5. void BGE (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si un dato es mayor o igual al otro.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.6. void BGT (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si un dato es mayor al otro.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.7. void BHI (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si un dato es mayor al otro (unsigned)

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.8. void BL (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** LR)

Función encargada de saltar sin condición y modificar LR.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	*LR

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.9. void BLE (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si un dato es menor o igual al otro.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.10. void BLS (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si un dato es menor o igual al otro (unsigned)

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.11. void BLT (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si un dato es menor al otro.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.12. void BLX (unsigned long ** PC, unsigned long posicionSiguiente, unsigned long ** LR)

Función encargada de ir a una posición específica sin condición y modificar LR.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	posicionSiguiente
<i>unsigned</i>	*LR

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.13. void BMI (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si el dato es negativo.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.14. void BNE (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si los datos no son iguales.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.15. void BPL (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si el dato es positivo.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.16. void BVC (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si no hay sobreflujo.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.17. void BVS (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si hay sobreflujo.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.1.1.18. void BX (unsigned long ** PC, unsigned long posicionSiguiente)

Función encargada de ir a una posición específica sin condición.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	posicionSiguiente

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2. Referencia del Archivo branch.h

Funciones

- void **B** (unsigned long **PC, unsigned long numeroSaltos)
Función encargada de saltar sin condición.
- void **BEQ** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si los datos son iguales.
- void **BNE** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si los datos no son iguales.
- void **BCS** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si un dato es mayor o igual al otro (unsigned)
- void **BCC** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si un dato es menor al otro (unsigned)
- void **BMI** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)
Función encargada de saltar si el dato es negativo.
- void **BPL** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)

Función encargada de saltar si el dato es positivo.

- void **BVS** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)

Función encargada de saltar si hay sobreflujo.

- void **BVC** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)

Función encargada de saltar si no hay sobreflujo.

- void **BHI** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)

Función encargada de saltar si un dato es mayor al otro (unsigned)

- void **BLS** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)

Función encargada de saltar si un dato es menor o igual al otro (unsigned)

- void **BGE** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)

Función encargada de saltar si un dato es mayor o igual al otro.

- void **BLT** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)

Función encargada de saltar si un dato es menor al otro.

- void **BGT** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)

Función encargada de saltar si un dato es mayor al otro.

- void **BLE** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **bandera)

Función encargada de saltar si un dato es menor o igual al otro.

- void **BL** (unsigned long **PC, unsigned long numeroSaltos, unsigned long **LR)

Función encargada de saltar sin condición y modificar LR.

- void **BX** (unsigned long **PC, unsigned long posicionSiguiente)

Función encargada de ir a una posición específica sin condición.

- void **BLX** (unsigned long **PC, unsigned long posicionSiguiente, unsigned long **LR)

Función encargada de ir a una posición específica sin condición y modificar LR.

5.2.1. Documentación de las funciones

5.2.1.1. void B (unsigned long ** PC, unsigned long numeroSaltos)

Función encargada de saltar sin condición.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.2. void BCC (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si un dato es menor al otro (unsigned)

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.3. void BCS (unsigned long ** *PC*, unsigned long *numeroSaltos*, unsigned long ** *bandera*)

Función encargada de saltar si un dato es mayor o igual al otro (unsigned)

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.4. void BEQ (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si los datos son iguales.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.5. void BGE (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si un dato es mayor o igual al otro.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.6. void BGT (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si un dato es mayor al otro.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.7. void BHI (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si un dato es mayor al otro (unsigned)

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.8. void BL (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** LR)

Función encargada de saltar sin condición y modificar LR.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	*LR

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.9. void BLE (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si un dato es menor o igual al otro.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.10. void BLS (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si un dato es menor o igual al otro (unsigned)

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.11. void BLT (unsigned long ** PC, unsigned long numeroSaltos, unsigned long ** bandera)

Función encargada de saltar si un dato es menor al otro.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.12. void BLX (*unsigned long* ** PC, *unsigned long* posicionSiguiente, *unsigned long* ** LR)

Función encargada de ir a una posición específica sin condición y modificar LR.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	posicionSiguiente
<i>unsigned</i>	*LR

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.13. void BMI (*unsigned long* ** PC, *unsigned long* numeroSaltos, *unsigned long* ** bandera)

Función encargada de saltar si el dato es negativo.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.14. void BNE (*unsigned long* ** PC, *unsigned long* numeroSaltos, *unsigned long* ** bandera)

Función encargada de saltar si los datos no son iguales.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.15. void BPL (*unsigned long* ** PC, *unsigned long* numeroSaltos, *unsigned long* ** bandera)

Función encargada de saltar si el dato es positivo.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.16. void BVC (unsigned long ** PC, unsigned long *numeroSaltos*, unsigned long ** *bandera*)

Función encargada de saltar si no hay sobreflujo.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.17. void BVS (unsigned long ** PC, unsigned long *numeroSaltos*, unsigned long ** *bandera*)

Función encargada de saltar si hay sobreflujo.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	numeroSaltos
<i>unsigned</i>	**bandera

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.2.1.18. void BX (unsigned long ** PC, unsigned long *posicionSiguiente*)

Función encargada de ir a una posición específica sin condición.

Parámetros

<i>unsigned</i>	**PC
<i>unsigned</i>	posicionSiguiente

Devuelve

Función tipo void pues modifica el valor del puntero PC

5.3. Referencia del Archivo decoder.c

```
#include "decoder.h"
#include <stdio.h>
#include <stdlib.h>
#include "registro.h"
#include "operaciones_alu.h"
#include <curses.h>
#include "branch.h"
#include "instrucciones_desplazamiento.h"
```

Funciones

- void [decodeInstruction](#) ([instruction_t](#) instruction, unsigned long *r[], unsigned long *bandera, unsigned long *PC, unsigned long *LR)
- [instruction_t](#) [getInstruction](#) (char *instStr)
- 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*, unsigned long * *r*[], unsigned long * *bandera*, unsigned long * *PC*, unsigned long * *LR*)

5.3.1.3. [instruction_t](#) [getInstruction](#) (char * *instStr*)

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, unsigned long *r[], unsigned long *bandera, unsigned long *PC, unsigned long *LR)
- [instruction_t](#) [getInstruction](#) (char *instStr)

- 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*, unsigned long * *r[]*, unsigned long * *bandera*, unsigned long * *PC*, unsigned long * *LR*)

5.4.1.3. `instruction_t` `getInstruction` (char * *instStr*)

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

5.5. Referencia del Archivo funciones_pila.c

```
#include "funciones_pila.h"
```

Funciones

- void `PUSH` (unsigned long registros[16], unsigned long *R, unsigned long *memoria)
Funcion encargada de pasar datos registro-memoria.
- void `POP` (unsigned long registros[16], unsigned long *R, unsigned long *memoria)
Funcion encargada de pasar datos memoria-registro.

5.5.1. Documentación de las funciones

5.5.1.1. void `POP` (unsigned long *registros[16]*, unsigned long * *R*, unsigned long * *memoria*)

Funcion encargada de pasar datos memoria-registro.

Parámetros

<i>unsigned</i>	long registros[16]
<i>unsigned</i>	long *R
<i>unsigned</i>	long *memoria

Devuelve

Funcion tipo void pues modifica directamente el valor de los registros

5.5.1.2. void `PUSH` (unsigned long *registros[16]*, unsigned long * *R*, unsigned long * *memoria*)

Funcion encargada de pasar datos registro-memoria.

Parámetros

<i>unsigned</i>	long registros[16]
<i>unsigned</i>	long *R

<i>unsigned</i>	long *memoria
-----------------	---------------

Devuelve

Funcion tipo void pues modifica directamente el valor de la memoria

5.6. Referencia del Archivo funciones_pila.h

Funciones

- void **PUSH** (unsigned long registros[16], unsigned long *R, unsigned long *memoria)
Funcion encargada de pasar datos registro-memoria.
- void **POP** (unsigned long registros[16], unsigned long *R, unsigned long *memoria)
Funcion encargada de pasar datos memoria-registro.

5.6.1. Documentación de las funciones

5.6.1.1. void POP (unsigned long *registros*[16], unsigned long * *R*, unsigned long * *memoria*)

Funcion encargada de pasar datos memoria-registro.

Parámetros

<i>unsigned</i>	long registros[16]
<i>unsigned</i>	long *R
<i>unsigned</i>	long *memoria

Devuelve

Funcion tipo void pues modifica directamente el valor de los registros

5.6.1.2. void PUSH (unsigned long *registros*[16], unsigned long * *R*, unsigned long * *memoria*)

Funcion encargada de pasar datos registro-memoria.

Parámetros

<i>unsigned</i>	long registros[16]
<i>unsigned</i>	long *R
<i>unsigned</i>	long *memoria

Devuelve

Funcion tipo void pues modifica directamente el valor de la memoria

5.7. Referencia del Archivo instrucciones_desplazamiento.c

```
#include "instrucciones_desplazamiento.h"
```

Funciones

- void **NOP** ()
Función que genera un retardo.
- unsigned long **LSL** (unsigned long Variable_entrada, unsigned long constante_parametro)
Función de desplazamiento a la izquierda.
- unsigned long **LSR** (unsigned long Variable_entrada, unsigned long constante_parametro)
Función de desplazamiento a la derecha.
- unsigned long **ROR** (unsigned long Variable_entrada, unsigned long constante_parametro)
Función de rotación a la derecha.
- unsigned long **ASR** (unsigned long Variable_entrada, unsigned long constante_parametro)
Función de desplazamiento aritmetico a la derecha.
- unsigned long **BIC** (unsigned long Variable_entrada)
Función encargada de multiplicar un dato por el negado de otro.
- unsigned long **MVN** (unsigned long Variable_entrada)
Función encargada de negar un número y almacenarlo en otra variable.
- unsigned long **RSBS** (unsigned long Variable_entrada, unsigned long constante_parametro)
Función encargada de volver negativo un número.
- unsigned long **REV** (unsigned long Variable_entrada)
Función encargada de cambiar la posición de los bits de un dato (grupos de 8 bits)
- unsigned long **REV16** (unsigned long Variable_entrada)
Función encargada de cambiar la posición de los bits de un dato (grupos de 16 bits)
- unsigned long **REVSH** (unsigned long Variable_entrada)
Función encargada de cambiar la posición de los dos primeros bytes de un datos y extender el signo.

5.7.1. Documentación de las funciones

5.7.1.1. unsigned long ASR (unsigned long Variable_entrada, unsigned long constante_parametro)

Función de desplazamiento aritmetico a la derecha.

Parámetros

<i>unsigned</i>	Variable_entrada
<i>unsigned</i>	constante_parametro

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.7.1.2. unsigned long BIC (unsigned long Variable_entrada)

Función encargada de multiplicar un dato por el negado de otro.

Parámetros

<i>unsigned</i>	Variable_entrada
-----------------	------------------

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.7.1.3. unsigned long LSL (unsigned long Variable_entrada, unsigned long constante_parametro)

Función de desplazamiento a la izquierda.

Parámetros

<i>unsigned</i>	Variable_entrada
<i>unsigned</i>	constante_parametro

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.7.1.4. unsigned long LSR (unsigned long *Variable_entrada*, unsigned long *constante_parametro*)

Función de desplazamiento a la derecha.

Parámetros

<i>unsigned</i>	Variable_entrada
<i>unsigned</i>	constante_parametro

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.7.1.5. unsigned long MVN (unsigned long *Variable_entrada*)

Función encargada de negar un número y almacenarlo en otra variable.

Parámetros

<i>unsigned</i>	Variable_entrada
-----------------	------------------

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.7.1.6. void NOP ()

Función que genera un retardo.

Devuelve

void

5.7.1.7. unsigned long REV (unsigned long *Variable_entrada*)

Función encargada de cambiar la posición de los bits de un dato (grupos de 8 bits)

Parámetros

<i>unsigned</i>	Variable_entrada
-----------------	------------------

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.7.1.8. unsigned long REV16 (unsigned long *Variable_entrada*)

Función encargada de cambiar la posición de los bits de un dato (grupos de 16 bits)

Parámetros

<i>unsigned</i>	Variable_entrada
-----------------	------------------

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.7.1.9. unsigned long REVSH (unsigned long *variable_entrada*)

Función encargada de cambiar la posición de los dos primeros bytes de un datos y extender el signo.

Parámetros

<i>unsigned</i>	Variable_entrada
-----------------	------------------

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.7.1.10. unsigned long ROR (unsigned long *Variable_entrada*, unsigned long *constante_parametro*)

Función de rotación a la derecha.

Parámetros

<i>unsigned</i>	Variable_entrada
<i>unsigned</i>	constante_parametro

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.7.1.11. unsigned long RSBS (unsigned long *Variable_entrada*, unsigned long *constante_parametro*)

Función encargada de volver negativo un número.

Parámetros

<i>unsigned</i>	Variable_entrada
<i>unsigned</i>	constante_parametro

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.8. Referencia del Archivo instrucciones_desplazamiento.h**Funciones**

- void **NOP** ()
Función que genera un retardo.
- unsigned long **LSL** (unsigned long Variable_entrada, unsigned long constante_parametro)
Función de desplazamiento a la izquierda.
- unsigned long **LSR** (unsigned long Variable_entrada, unsigned long constante_parametro)

Función de desplazamiento a la derecha.

- unsigned long **ROR** (unsigned long Variable_entrada, unsigned long constante_parametro)

Función de rotación a la derecha.

- unsigned long **ASR** (unsigned long Variable_entrada, unsigned long constante_parametro)

Función de desplazamiento aritmetico a la derecha.

- unsigned long **BIC** (unsigned long Variable_entrada)

Función encargada de multiplicar un dato por el negado de otro.

- unsigned long **MVN** (unsigned long Variable_entrada)

Función encargada de negar un número y almacenarlo en otra variable.

- unsigned long **RSBS** (unsigned long Variable_entrada, unsigned long constante_parametro)

Función encargada de volver negativo un número.

- unsigned long **REV** (unsigned long Variable_entrada)

Función encargada de cambiar la posición de los bits de un dato (grupos de 8 bits)

- unsigned long **REV16** (unsigned long Variable_entrada)

Función encargada de cambiar la posición de los bits de un dato (grupos de 16 bits)

- unsigned long **REVSH** (unsigned long variable_entrada)

Función encargada de cambiar la posición de los dos primeros bytes de un datos y extender el signo.

5.8.1. Documentación de las funciones

5.8.1.1. unsigned long ASR (unsigned long Variable_entrada, unsigned long constante_parametro)

Función de desplazamiento aritmetico a la derecha.

Parámetros

<i>unsigned</i>	Variable_entrada
<i>unsigned</i>	constante_parametro

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.8.1.2. unsigned long BIC (unsigned long Variable_entrada)

Función encargada de multiplicar un dato por el negado de otro.

Parámetros

<i>unsigned</i>	Variable_entrada
-----------------	------------------

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.8.1.3. unsigned long LSL (unsigned long Variable_entrada, unsigned long constante_parametro)

Función de desplazamiento a la izquierda.

Parámetros

<i>unsigned</i>	Variable_entrada
<i>unsigned</i>	constante_parametro

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.8.1.4. unsigned long LSR (unsigned long Variable_entrada, unsigned long constante_parametro)

Función de desplazamiento a la derecha.

Parámetros

<i>unsigned</i>	Variable_entrada
<i>unsigned</i>	constante_parametro

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.8.1.5. unsigned long MVN (unsigned long Variable_entrada)

Función encargada de negar un número y almacenarlo en otra variable.

Parámetros

<i>unsigned</i>	Variable_entrada
-----------------	------------------

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.8.1.6. void NOP ()

Función que genera un retardo.

Devuelve

void

5.8.1.7. unsigned long REV (unsigned long Variable_entrada)

Función encargada de cambiar la posición de los bits de un dato (grupos de 8 bits)

Parámetros

<i>unsigned</i>	Variable_entrada
-----------------	------------------

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.8.1.8. unsigned long REV16 (unsigned long Variable_entrada)

Función encargada de cambiar la posición de los bits de un dato (grupos de 16 bits)

Parámetros

<i>unsigned</i>	Variable_entrada
-----------------	------------------

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.8.1.9. unsigned long REVSH (unsigned long *variable_entrada*)

Función encargada de cambiar la posición de los dos primeros bytes de un datos y extender el signo.

Parámetros

<i>unsigned</i>	Variable_entrada
-----------------	------------------

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.8.1.10. unsigned long ROR (unsigned long *Variable_entrada*, unsigned long *constante_parametro*)

Función de rotación a la derecha.

Parámetros

<i>unsigned</i>	Variable_entrada
<i>unsigned</i>	constante_parametro

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.8.1.11. unsigned long RSBS (unsigned long *Variable_entrada*, unsigned long *constante_parametro*)

Función encargada de volver negativo un número.

Parámetros

<i>unsigned</i>	Variable_entrada
<i>unsigned</i>	constante_parametro

Devuelve

Un registro de tipo long sin signo con la instrucción realizada

5.9. Referencia del Archivo mostrar_resultado.c

```
#include <stdio.h>
#include <stdlib.h>
#include "registro.h"
#include "operaciones_alu.h"
#include "mostrar_resultado.h"
#include <curses.h>
```

Funciones

- void **mostrar** (unsigned long **resultado**)

Función encargada mostrar el resultado en una sección de la interfaz.

5.9.1. Documentación de las funciones

5.9.1.1. void mostrar (unsigned long **resultado**)

Función encargada mostrar el resultado en una sección de la interfaz.

Parámetros

<i>unsigned</i>	long resultado
-----------------	----------------

Devuelve

Función declarada tipo void pues solo imprime datos

5.10. Referencia del Archivo **mostrar_resultado.h**

Funciones

- void **mostrar** (unsigned long **resultado**)

Función encargada mostrar el resultado en una sección de la interfaz.

5.10.1. Documentación de las funciones

5.10.1.1. void mostrar (unsigned long **resultado**)

Función encargada mostrar el resultado en una sección de la interfaz.

Parámetros

<i>unsigned</i>	long resultado
-----------------	----------------

Devuelve

Función declarada tipo void pues solo imprime datos

5.11. Referencia del Archivo **operaciones_alu.c**

```
#include <stdio.h>
#include "operaciones_alu.h"
```

Funciones

- unsigned long **ADD** (unsigned long op1, unsigned long op2, unsigned long **bandera)
- unsigned long **ADC** (unsigned long op1, unsigned long op2, unsigned long **bandera)

Función encargada de sumar teniendo en cuenta el carry.

- unsigned long **AND** (unsigned long op1, unsigned long op2, unsigned long **bandera)

Función encargada de realizar una multiplicación lógica.

- unsigned long **EOR** (unsigned long op1, unsigned long op2, unsigned long **bandera)
Función encargada de realizar una EX-OR lógica.
- unsigned **MOV** (unsigned long op1, unsigned long op2, unsigned long **bandera)
Función encargada copiar un dato en una variable diferente.
- unsigned long **ORR** (unsigned long op1, unsigned long op2, unsigned long **bandera)
Función encargada de realizar una OR lógica.
- unsigned long **SUB** (unsigned long op1, unsigned long op2, unsigned long **bandera)
Función encargada restar.
- void **BDES** (unsigned long op1, unsigned long **bandera)
Función encargada de modificar las banderas de las funciones desplazamiento.

Variables

- unsigned long **constante** =2147483647
- unsigned long **resultado**
- int **auxban**

5.11.1. Documentación de las funciones

5.11.1.1. unsigned long ADC (unsigned long op1, unsigned long op2, unsigned long ** bandera)

Función encargada de sumar teniendo en cuenta el carry.

Parámetros

<i>unsigned</i>	long op1
<i>unsigned</i>	long op2
<i>unsigned</i>	long **bandera

Devuelve

Un dato tipo long sin signo con el resultado de la operación

5.11.1.2. unsigned long ADD (unsigned long op1, unsigned long op2, unsigned long ** bandera)

5.11.1.3. unsigned long AND (unsigned long op1, unsigned long op2, unsigned long ** bandera)

Función encargada de realizar una multiplicación lógica.

Parámetros

<i>unsigned</i>	long op1
<i>unsigned</i>	long op2
<i>unsigned</i>	long **bandera

Devuelve

Un dato tipo long sin signo con el resultado de la operación

5.11.1.4. void BDES (unsigned long op1, unsigned long ** bandera)

Función encargada de modificar las banderas de las funciones desplazamiento.

Parámetros

<i>unsigned</i>	long op1
<i>unsigned</i>	long **bandera

Devuelve

nada

5.11.1.5. unsigned long EOR (unsigned long op1, unsigned long op2, unsigned long ** bandera)

Función encargada de realizar una EX-OR lógica.

Parámetros

<i>unsigned</i>	long op1
<i>unsigned</i>	long op2
<i>unsigned</i>	long **bandera

Devuelve

Un dato tipo long sin signo con el resultado de la operación

5.11.1.6. unsigned MOV (unsigned long op1, unsigned long op2, unsigned long ** bandera)

Función encargada copiar un dato en una variable diferente.

Parámetros

<i>unsigned</i>	long op1
<i>unsigned</i>	long op2
<i>unsigned</i>	long **bandera

Devuelve

Un dato tipo long sin signo con el resultado de la operación

5.11.1.7. unsigned long ORR (unsigned long op1, unsigned long op2, unsigned long ** bandera)

Función encargada de realizar una OR lógica.

Parámetros

<i>unsigned</i>	long op1
<i>unsigned</i>	long op2
<i>unsigned</i>	long **bandera

Devuelve

Un dato tipo long sin signo con el resultado de la operación

5.11.1.8. unsigned long SUB (unsigned long op1, unsigned long op2, unsigned long ** bandera)

Función encargada restar.

Parámetros

<i>unsigned</i>	long op1
<i>unsigned</i>	long op2
<i>unsigned</i>	long **bandera

Devuelve

Un dato tipo long sin signo con el resultado de la operación

5.11.2. Documentación de las variables

5.11.2.1. int auxban

5.11.2.2. unsigned long constante =2147483647

5.11.2.3. unsigned long resultado

5.12. Referencia del Archivo operaciones_alu.h

Funciones

- *****brief** Función encargada sumar *param unsigned long op1 *param unsigned long op2 *param unsigned long **bandera *return Un dato tipo long sin signo con el **resultado** de la operación *unsigned long **ADD** (unsigned long op1, unsigned long op2, unsigned long **bandera)
- unsigned long **AND** (unsigned long op1, unsigned long op2, unsigned long **bandera)
Función encargada de realizar una multiplicación lógica.
- unsigned long **EOR** (unsigned long op1, unsigned long op2, unsigned long **bandera)
Función encargada de realizar una EX-OR lógica.
- unsigned **MOV** (unsigned long op1, unsigned long op2, unsigned long **bandera)
Función encargada copiar un dato en una variable diferente.
- unsigned long **ORR** (unsigned long op1, unsigned long op2, unsigned long **bandera)
Función encargada de realizar una OR lógica.
- unsigned long **SUB** (unsigned long op1, unsigned long op2, unsigned long **bandera)
Función encargada restar.
- unsigned long **ADC** (unsigned long op1, unsigned long op2, unsigned long **bandera)
Función encargada de sumar teniendo en cuenta el carry.
- void **BDES** (unsigned long op1, unsigned long **bandera)
Función encargada de modificar las banderas de las funciones desplazamiento.

5.12.1. Documentación de las funciones

5.12.1.1. unsigned long ADC (unsigned long op1, unsigned long op2, unsigned long ** bandera)

Función encargada de sumar teniendo en cuenta el carry.

Parámetros

<i>unsigned</i>	long op1
<i>unsigned</i>	long op2

<i>unsigned</i>	long **bandera
-----------------	----------------

Devuelve

Un dato tipo long sin signo con el resultado de la operación

5.12.1.2. **** * brief** Función encargada sumar* param unsigned long op1* param unsigned long op2* param unsigned long** bandera* return Un dato tipo long sin signo con el resultado de la operación* unsigned long ADD (unsigned long op1, unsigned long op2, unsigned long ** bandera)

5.12.1.3. unsigned long AND (unsigned long op1, unsigned long op2, unsigned long ** bandera)

Función encargada de realizar una multiplicación lógica.

Parámetros

<i>unsigned</i>	long op1
<i>unsigned</i>	long op2
<i>unsigned</i>	long **bandera

Devuelve

Un dato tipo long sin signo con el resultado de la operación

5.12.1.4. void BDES (unsigned long op1, unsigned long ** bandera)

Función encargada de modificar las banderas de las funciones desplazamiento.

Parámetros

<i>unsigned</i>	long op1
<i>unsigned</i>	long **bandera

Devuelve

nada

5.12.1.5. unsigned long EOR (unsigned long op1, unsigned long op2, unsigned long ** bandera)

Función encargada de realizar una EX-OR lógica.

Parámetros

<i>unsigned</i>	long op1
<i>unsigned</i>	long op2
<i>unsigned</i>	long **bandera

Devuelve

Un dato tipo long sin signo con el resultado de la operación

5.12.1.6. unsigned MOV (unsigned long op1, unsigned long op2, unsigned long ** bandera)

Función encargada copiar un dato en una variable diferente.

Parámetros

<i>unsigned</i>	long op1
<i>unsigned</i>	long op2
<i>unsigned</i>	long **bandera

Devuelve

Un dato tipo long sin signo con el resultado de la operación

5.12.1.7. unsigned long ORR (unsigned long op1, unsigned long op2, unsigned long ** bandera)

Función encargada de realizar una OR lógica.

Parámetros

<i>unsigned</i>	long op1
<i>unsigned</i>	long op2
<i>unsigned</i>	long **bandera

Devuelve

Un dato tipo long sin signo con el resultado de la operación

5.12.1.8. unsigned long SUB (unsigned long op1, unsigned long op2, unsigned long ** bandera)

Función encargada restar.

Parámetros

<i>unsigned</i>	long op1
<i>unsigned</i>	long op2
<i>unsigned</i>	long **bandera

Devuelve

Un dato tipo long sin signo con el resultado de la operación

5.13. Referencia del Archivo README.md**5.14. Referencia del Archivo registro.c**

```
#include <stdio.h>
#include <stdlib.h>
#include "registro.h"
#include <curses.h>
```

Funciones

- void [mostrar_registros](#) (unsigned long r[13])

Función para mostrar los datos almacenados en los registros.

5.14.1. Documentación de las funciones

5.14.1.1. void mostrar_registros (unsigned long *r*[13])

Función para mostrar los datos almacenados en los registros.

Parámetros

<i>unsigned</i>	long r[13]
-----------------	------------

Devuelve

Función declarada tipo void, pues solamente muestra datos en pantalla

5.15. Referencia del Archivo registro.h

Funciones

- void `mostrar_registros` (unsigned long r[13])
Función para mostrar los datos almacenados en los registros.

5.15.1. Documentación de las funciones

5.15.1.1. void `mostrar_registros` (unsigned long r[13])

Función para mostrar los datos almacenados en los registros.

Parámetros

<i>unsigned</i>	long r[13]
-----------------	------------

Devuelve

Función declarada tipo void, pues solamente muestra datos en pantalla

5.16. Referencia del Archivo test.c

```
#include <stdio.h>
#include <stdlib.h>
#include "registro.h"
#include "operaciones_alu.h"
#include <curses.h>
#include "decoder.h"
```

Funciones

- int `main` ()

5.16.1. Documentación de las funciones

5.16.1.1. int `main` ()

