Emulador ARM Cortex M0

Generated by Doxygen 1.8.10

Sat Sep 19 2015 15:30:42

Contents

Chapter 1

Practica #1 Emulador ARM Cortex -M0

Documentacion pertinente del sofware realizado para realizar el Emulador del procesador ARM Cortex -M0

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Data Structure Documentation

4.1 u32tobyte_t Union Reference

Data Fields

- uint32_t data
- struct {
    uint8_t byte0
    uint8_t byte1
    uint8_t byte2
    uint8_t byte3
  };

4.1.1 Field Documentation

4.1.1.1 struct { ... }

4.1.1.2 uint8_t byte0

4.1.1.3 uint8_t byte1

4.1.1.4 uint8_t byte2

4.1.1.5 uint8_t byte3

4.1.1.6 uint32_t data

The documentation for this union was generated from the following file:

- Instrucciones_desplazamiento.c

Chapter 5

File Documentation

5.1 Banderas.c File Reference

Contiene las funciones para la correcta activacion de las banderas N,Z,C,V.

```
#include "Banderas.h"
```

Functions

- void BANDERAS (uint32_t Rd, uint32_t Rn, uint32_t Rm, int *Banderas)

  Funcion donde se llevara a cabo la activacion de las banderas.
- void BANDERAS_DES (uint32_t Rd, int *Banderas)

  Funcion donde se llevara a cabo la activacion de las banderas.

5.1.1 Detailed Description

Contiene las funciones para la correcta activacion de las banderas N,Z,C,V.

5.1.2 Function Documentation

5.1.2.1 void BANDERAS (uint32_t *Rd,* uint32_t *Rn,* uint32_t *Rm,* int * *Banderas*)

Funcion donde se llevara a cabo la activacion de las banderas.

Parameters

Rd	Resultado
Rn	Primer dato de la operacion realizada
Rm	Segundo dato de la operacion realizada
Banderas	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.1.2.2 void BANDERAS_DES (uint32_t *Rd,* int * *Banderas*)

Funcion donde se llevara a cabo la activacion de las banderas.

Parameters

Rd	Resultado
Banderas	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.2 Banderas.h File Reference

Archivo que contiene las definiciones de las funciones para la activacion de las bandera.

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

Functions

- void BANDERAS (uint32_t Rd, uint32_t Rn, uint32_t Rm, int *Banderas)

  Funcion donde se llevara a cabo la activacion de las banderas.
- void BANDERAS_DES (uint32_t Rd, int *Banderas)

  Funcion donde se llevara a cabo la activacion de las banderas.

5.2.1 Detailed Description

Archivo que contiene las definiciones de las funciones para la activacion de las bandera.

5.2.2 Function Documentation

5.2.2.1 void BANDERAS (uint32_t *Rd,* uint32_t *Rn,* uint32_t *Rm,* int * *Banderas*)

Funcion donde se llevara a cabo la activacion de las banderas.

Parameters

Rd	Resultado
Rn	Primer dato de la operacion realizada
Rm	Segundo dato de la operacion realizada
Banderas	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.2.2.2 void BANDERAS_DES (uint32_t *Rd,* int * *Banderas*)

Funcion donde se llevara a cabo la activacion de las banderas.

Parameters

Rd	Resultado
Banderas	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.3 Instrucciones.c File Reference

libreria encargada de realizar las funciones aritmeticas

```
#include "Instrucciones.h"
#include "Banderas.h"
```

Functions

- void ADDS (uint32_t *Rd, uint32_t Rn, uint32_t Rm, int *flags)

  Funcion suma.
- void ANDS (uint32_t *Rd, uint32_t Rn, uint32_t Rm, int *flags)

  Funcion logica AND, bit a bit.
- void EORS (uint32_t *Rd, uint32_t Rn, uint32_t Rm, int *flags)

  Funcion logica X-OR, bit a bit.
- void MOVS (uint32_t *Rd, uint32_t Rn)

  Funcion realiza una copia de un registro en otro.
- void ORRS (uint32_t *Rd, uint32_t Rn, uint32_t Rm, int *flags)

  Funcion logica OR, bit a bit.
- void SUBS (uint32_t *Rd, uint32_t Rn, uint32_t Rm, int *flags)

  Funcion resta.
- void CMNS (uint32_t Rn, uint32_t Rm, int *flags)

  Funcion suma,pero solo modifica las banderas.
- void CMPS (uint32_t Rn, uint32_t Rm, int *flags)

  Funcion resta, pero solo modifica las banderas.
- void MULS (uint32_t *Rd, uint32_t Rn, uint32_t Rm, int *flags)

  Funcion multiplica, guarda 32 bits menos significativos.
- void TST (uint32_t Rn, uint32_t Rm, int *flags)

  Funcion logica AND, bit a bit pero solo modifica las banderas.
- void NOP ()

  Funcion que no realiza ninguna operacion.
- void ADCS (uint32_t *Rd, uint32_t Rn, uint32_t Rm, int *flags)

  Funcion suma con acarreo.
- void SBCS (uint32_t *Rd, uint32_t Rn, uint32_t Rm, int *flags)

  Funcion resta con acarreo.

5.3.1 Detailed Description

libreria encargada de realizar las funciones aritmeticas

5.3.2 Function Documentation

5.3.2.1 void ADCS (uint32_t * *Rd,* uint32_t *Rn,* uint32_t *Rm,* int * *flags*)

Funcion suma con acarreo.

Parameters

∗Rd	Dato donde se almacenara el resultado
Rn	Primer dato a realizar la operacion
Rm	Segundo dato a realizar la operacion
∗flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.3.2.2 void ADDS (uint32_t ∗ *Rd*, uint32_t *Rn*, uint32_t *Rm*, int ∗ *flags*)

Funcion suma.

Parameters

∗Rd	Dato donde se almacenara el resultado
Rn	Primer dato a realizar la operacion
Rm	Segundo dato a realizar la operacion
∗flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.3.2.3 void ANDS (uint32_t ∗ *Rd*, uint32_t *Rn*, uint32_t *Rm*, int ∗ *flags*)

Funcion logica AND, bit a bit.

Parameters

∗Rd	Dato donde se almacenara el resultado
Rn	Primer dato a realizar la operacion
Rm	Primer dato a realizar la operacion
∗flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.3.2.4 void CMNS (uint32_t *Rn*, uint32_t *Rm*, int ∗ *flags*)

Funcion suma,pero solo modifica las banderas.

Parameters

Rn	Primer dato a realizar la operacion
Rm	Segundo dato a realizar la operacion
∗flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.3.2.5 void CMPS (uint32_t *Rn*, uint32_t *Rm*, int ∗ *flags*)

Funcion resta, pero solo modifica las banderas.

Parameters

Rn	Primer dato a realizar la operacion
Rm	Segundo dato a realizar la operacion
∗*flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.3.2.6 void EORS (uint32_t ∗ *Rd,* uint32_t *Rn,* uint32_t *Rm,* int ∗ *flags*)

Funcion logica X-OR, bit a bit.

Parameters

∗*Rd*	Dato donde se almacenara el resultado
Rn	Primer registro a relizar operacion
Rm	Primer registro a relizar operacion
∗*flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.3.2.7 void MOVS (uint32_t ∗ *Rd,* uint32_t *Rn*)

Funcion realiza una copia de un registro en otro.

Parameters

∗*Rd*	Dato donde se realizara la copia
Rn	Dato al que se le realizara una copia

Returns

No hay retorno de la funcion

5.3.2.8 void MULS (uint32_t ∗ *Rd,* uint32_t *Rn,* uint32_t *Rm,* int ∗ *flags*)

Funcion multiplica, guarda 32 bits menos significativos.

Parameters

∗*Rd*	Dato donde se almacenara el resultado
Rn	Primer dato a realizar la operacion
Rm	Segundo dato a realizar la operacion
∗*flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.3.2.9 void NOP ( )

Funcion que no realiza ninguna operacion.

Returns

No hay retorno de la funcion

5.3.2.10 void ORRS (uint32_t * *Rd*, uint32_t *Rn*, uint32_t *Rm*, int * *flags*)

Funcion logica OR, bit a bit.

Parameters

**Rd*	Dato donde se almacenara el resultado
Rn	Primer registro a relizar operacion
Rm	Primer registro a relizar operacion
**flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.3.2.11 void SBCS (uint32_t * *Rd*, uint32_t *Rn*, uint32_t *Rm*, int * *flags*)

Funcion resta con acarreo.

Parameters

**Rd*	Dato donde se almacenara el resultado
Rn	Primer dato a realizar la operacion
Rm	Segundo dato a realizar la operacion
**flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.3.2.12 void SUBS (uint32_t * *Rd*, uint32_t *Rn*, uint32_t *Rm*, int * *flags*)

Funcion resta.

Parameters

**Rd*	Dato donde se almacenara el resultado
Rn	Primer dato a realizar la operacion
Rm	Segundo dato a realizar la operacion
**flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.3.2.13 void TST (uint32_t *Rn*, uint32_t *Rm*, int * *flags*)

Funcion logica AND, bit a bit pero solo modifica las banderas.

Parameters

Rn	Primer dato a realizar la operacion
Rm	Segundo dato a realizar la operacion
∗*flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.4 Instrucciones.h File Reference

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

Functions

- void ADDS (uint32_t ∗Rd, uint32_t Rn, uint32_t Rm, int ∗flags)
  Funcion suma.
- void ANDS (uint32_t ∗Rd, uint32_t Rn, uint32_t Rm, int ∗flags)
  Funcion logica AND, bit a bit.
- void EORS (uint32_t ∗Rd, uint32_t Rn, uint32_t Rm, int ∗flags)
  Funcion logica X-OR, bit a bit.
- void MOVS (uint32_t ∗Rd, uint32_t Rn)
  Funcion realiza una copia de un registro en otro.
- void ORRS (uint32_t ∗Rd, uint32_t Rn, uint32_t Rm, int ∗flags)
  Funcion logica OR, bit a bit.
- void SUBS (uint32_t ∗Rd, uint32_t Rn, uint32_t Rm, int ∗flags)
  Funcion resta.
- void CMNS (uint32_t Rn, uint32_t Rm, int ∗flags)
  Funcion suma,pero solo modifica las banderas.
- void CMPS (uint32_t Rn, uint32_t Rm, int ∗flags)
  Funcion resta, pero solo modifica las banderas.
- void MULS (uint32_t ∗Rd, uint32_t Rn, uint32_t Rm, int ∗flags)
  Funcion multiplica, guarda 32 bits menos significativos.
- void ADCS (uint32_t ∗Rd, uint32_t Rn, uint32_t Rm, int ∗flags)
  Funcion suma con acarreo.
- void SBCS (uint32_t ∗Rd, uint32_t Rn, uint32_t Rm, int ∗flags)
  Funcion resta con acarreo.
- void TST (uint32_t Rn, uint32_t Rm, int ∗flags)
  Funcion logica AND, bit a bit pero solo modifica las banderas.
- void NOP ()
  Funcion que no realiza ninguna operacion.

5.4.1 Function Documentation

5.4.1.1 void ADCS (uint32_t ∗ *Rd,* uint32_t *Rn,* uint32_t *Rm,* int ∗ *flags*)

Funcion suma con acarreo.

Parameters

∗Rd	Dato donde se almacenara el resultado
Rn	Primer dato a realizar la operacion
Rm	Segundo dato a realizar la operacion
∗flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.4.1.2 void ADDS (uint32_t ∗ *Rd,* uint32_t *Rn,* uint32_t *Rm,* int ∗ *flags*)

Funcion suma.

Parameters

∗Rd	Dato donde se almacenara el resultado
Rn	Primer dato a realizar la operacion
Rm	Segundo dato a realizar la operacion
∗flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.4.1.3 void ANDS (uint32_t ∗ *Rd,* uint32_t *Rn,* uint32_t *Rm,* int ∗ *flags*)

Funcion logica AND, bit a bit.

Parameters

∗Rd	Dato donde se almacenara el resultado
Rn	Primer dato a realizar la operacion
Rm	Primer dato a realizar la operacion
∗flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.4.1.4 void CMNS (uint32_t *Rn,* uint32_t *Rm,* int ∗ *flags*)

Funcion suma,pero solo modifica las banderas.

Parameters

Rn	Primer dato a realizar la operacion
Rm	Segundo dato a realizar la operacion
∗flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.4.1.5 void CMPS (uint32_t *Rn,* uint32_t *Rm,* int ∗ *flags*)

Funcion resta, pero solo modifica las banderas.

Parameters

Rn	Primer dato a realizar la operacion
Rm	Segundo dato a realizar la operacion
∗*flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.4.1.6 void EORS (uint32_t ∗ *Rd,* uint32_t *Rn,* uint32_t *Rm,* int ∗ *flags*)

Funcion logica X-OR, bit a bit.

Parameters

∗*Rd*	Dato donde se almacenara el resultado
Rn	Primer registro a relizar operacion
Rm	Primer registro a relizar operacion
∗*flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.4.1.7 void MOVS (uint32_t ∗ *Rd,* uint32_t *Rn*)

Funcion realiza una copia de un registro en otro.

Parameters

∗*Rd*	Dato donde se realizara la copia
Rn	Dato al que se le realizara una copia

Returns

No hay retorno de la funcion

5.4.1.8 void MULS (uint32_t ∗ *Rd,* uint32_t *Rn,* uint32_t *Rm,* int ∗ *flags*)

Funcion multiplica, guarda 32 bits menos significativos.

Parameters

∗*Rd*	Dato donde se almacenara el resultado
Rn	Primer dato a realizar la operacion
Rm	Segundo dato a realizar la operacion
∗*flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.4.1.9 void NOP ( )

Funcion que no realiza ninguna operacion.

Returns

No hay retorno de la funcion

5.4.1.10 void ORRS (uint32_t * *Rd*, uint32_t *Rn*, uint32_t *Rm*, int * *flags*)

Funcion logica OR, bit a bit.

Parameters

**Rd*	Dato donde se almacenara el resultado
Rn	Primer registro a relizar operacion
Rm	Primer registro a relizar operacion
**flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.4.1.11 void SBCS (uint32_t * *Rd*, uint32_t *Rn*, uint32_t *Rm*, int * *flags*)

Funcion resta con acarreo.

Parameters

**Rd*	Dato donde se almacenara el resultado
Rn	Primer dato a realizar la operacion
Rm	Segundo dato a realizar la operacion
**flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.4.1.12 void SUBS (uint32_t * *Rd*, uint32_t *Rn*, uint32_t *Rm*, int * *flags*)

Funcion resta.

Parameters

**Rd*	Dato donde se almacenara el resultado
Rn	Primer dato a realizar la operacion
Rm	Segundo dato a realizar la operacion
**flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.4.1.13 void TST (uint32_t *Rn*, uint32_t *Rm*, int * *flags*)

Funcion logica AND, bit a bit pero solo modifica las banderas.

Parameters

Rn	Primer dato a realizar la operacion
Rm	Segundo dato a realizar la operacion
∗*flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.5 Instrucciones_desplazamiento.c File Reference

libreria encargada de realizar las funciones de desplazamiento

```
#include "Instrucciones_desplazamiento.h"
#include "Banderas.h"
```

Data Structures

- union u32tobyte_t

Functions

- void LSLS (uint32_t ∗Rdn, uint32_t Rn, uint32_t Rm, int ∗flags)

  Desplazamiento logico a la izquierda.
- void LSRS (uint32_t ∗Rdn, uint32_t Rn, uint32_t Rm, int ∗flags)

  Desplazamiento logico a la izquierda.
- void ROR (uint32_t ∗Rdn, uint32_t Rm, int ∗flags)

  Rotacion a la derecha.
- void ASR (uint32_t ∗Rdn, uint32_t Rm, int ∗flags)

  Desplazamiento aritmetico a la derecha.
- void BIC (uint32_t ∗Rdn, uint32_t Rm, int ∗flags)

  Operacion logica AND entre un dato y el complemento del otro dato.
- void MVN (uint32_t ∗Rdn, uint32_t Rm, int ∗flags)

  Guardar el complemento de un dato.
- void RSB (uint32_t ∗Rdn, uint32_t Rm, int ∗flags)

  Complemento a dos de un dato.
- void REV (uint32_t ∗Rdn, uint32_t Rm, int ∗flags)

  Cambiar el orden de los bytes.
- void REV16 (uint32_t ∗Rdn, uint32_t Rm, int ∗flags)

  Cambiar el orden de los bytes en cada halfword de 16bits.
- void REVSH (uint32_t ∗Rdn, uint32_t Rm, int ∗flags)

  Cambiar el orden de los bytes del halfword bajo.

5.5.1 Detailed Description

libreria encargada de realizar las funciones de desplazamiento

5.5.2 Function Documentation

5.5.2.1 void ASR (uint32_t * *Rdn*, uint32_t *Rm*, int * *flags*)

Desplazamiento aritmetico a la derecha.

Parameters

∗Rdn	Dato que se desplazara
Rm	Numero de desplazamientos
∗flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.5.2.2 void BIC (uint32_t ∗ *Rdn,* uint32_t *Rm,* int ∗ *flags*)

Operacion logica AND entre un dato y el complemento del otro dato.

Parameters

∗Rdn	Primer operador de la operacion AND
Rm	Segundo operador de la operacion AND, al cual se le hara el complemento
∗flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.5.2.3 void LSLS (uint32_t ∗ *Rdn,* uint32_t *Rn,* uint32_t *Rm,* int ∗ *flags*)

Desplazamiento logico a la izquierda.

Parameters

∗Rdn	Dato donde se almacenara el resultado
Rn	Dato a desplazar
Rm	Numero de desplazamientos
∗flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.5.2.4 void LSRS (uint32_t ∗ *Rdn,* uint32_t *Rn,* uint32_t *Rm,* int ∗ *flags*)

Desplazamiento logico a la izquierda.

Parameters

∗Rdn	Dato donde se almacenara el resultado
Rn	Dato a desplazar
Rm	Numero de desplazamientos
∗flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.5.2.5 void MVN (uint32_t ∗ *Rdn,* uint32_t *Rm,* int ∗ *flags*)

Guardar el complemento de un dato.

Parameters

Rdn	Donde se almacenara el dato
Rm	Dato al que se le hara el complemento
flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.5.2.6 void REV (uint32_t * *Rdn*, uint32_t *Rm*, int * *flags*)

Cambiar el orden de los bytes.

Parameters

Rdn	Donde se almacenara el resultado
Rm	Dato que se le cambiara el orden de los bytes
flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.5.2.7 void REV16 (uint32_t * *Rdn*, uint32_t *Rm*, int * *flags*)

Cambiar el orden de los bytes en cada halfword de 16bits.

Parameters

Rdn	Donde se almacenara el resultado
Rm	Dato que se le cambiara el orden de los bytes
flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.5.2.8 void REVSH (uint32_t * *Rdn*, uint32_t *Rm*, int * *flags*)

Cambiar el orden de los bytes del halfword bajo.

Parameters

**Rdn*	Donde se almacenara el resultado
Rm	Dato que se le cambiara el orden de los bytes
**flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.5.2.9 void ROR (uint32_t * *Rdn*, uint32_t *Rm*, int * *flags*)

Rotacion a la derecha.

Parameters

∗Rdn	Dato que se rotara
Rm	Orden de la rotacion

Returns

No hay retorno de la funcion

5.5.2.10 void RSB (uint32_t ∗ *Rdn*, uint32_t *Rm*, int ∗ *flags*)

Complemento a dos de un dato.

Parameters

Rdn	Donde se almacenara el dato
Rm	Dato al que se le hara el complemento a dos
flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.6 Instrucciones_desplazamiento.h File Reference

Archivo que contiene las definiciones de las funciones de desplazamiento.

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

Functions

- void LSLS (uint32_t ∗Rdn, uint32_t Rn, uint32_t Rm, int ∗flags)
  Desplazamiento logico a la izquierda.
- void LSRS (uint32_t ∗Rdn, uint32_t Rn, uint32_t Rm, int ∗flags)
  Desplazamiento logico a la izquierda.
- void ROR (uint32_t ∗Rdn, uint32_t Rm, int ∗flags)
  Rotacion a la derecha.
- void ASR (uint32_t ∗Rdn, uint32_t Rm, int ∗flags)
  Desplazamiento aritmetico a la derecha.
- void BIC (uint32_t ∗Rdn, uint32_t Rm, int ∗flags)
  Operacion logica AND entre un dato y el complemento del otro dato.
- void MVN (uint32_t ∗Rdn, uint32_t Rm, int ∗flags)
  Guardar el complemento de un dato.
- void RSB (uint32_t ∗Rdn, uint32_t Rm, int ∗flags)
  Complemento a dos de un dato.
- void REV (uint32_t ∗Rdn, uint32_t Rm, int ∗flags)
  Cambiar el orden de los bytes.
- void REV16 (uint32_t ∗Rdn, uint32_t Rm, int ∗flags)
  Cambiar el orden de los bytes en cada halfword de 16bits.
- void REVSH (uint32_t ∗Rdn, uint32_t Rm, int ∗flags)
  Cambiar el orden de los bytes del halfword bajo.

5.6.1 Detailed Description

Archivo que contiene las definiciones de las funciones de desplazamiento.

5.6.2 Function Documentation

5.6.2.1 void ASR (uint32_t * *Rdn*, uint32_t *Rm*, int * *flags*)

Desplazamiento aritmetico a la derecha.

Parameters

**Rdn*	Dato que se desplazara
Rm	Numero de desplazamientos
**flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.6.2.2 void BIC (uint32_t * *Rdn*, uint32_t *Rm*, int * *flags*)

Operacion logica AND entre un dato y el complemento del otro dato.

Parameters

**Rdn*	Primer operador de la operacion AND
Rm	Segundo operador de la operacion AND, al cual se le hara el complemento
**flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.6.2.3 void LSLS (uint32_t * *Rdn*, uint32_t *Rn*, uint32_t *Rm*, int * *flags*)

Desplazamiento logico a la izquierda.

Parameters

**Rdn*	Dato donde se almacenara el resultado
Rn	Dato a desplazar
Rm	Numero de desplazamientos
**flags*	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.6.2.4 void LSRS (uint32_t * *Rdn*, uint32_t *Rn*, uint32_t *Rm*, int * *flags*)

Desplazamiento logico a la izquierda.

Parameters

∗Rdn	Dato donde se almacenara el resultado
Rn	Dato a desplazar
Rm	Numero de desplazamientos
∗flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.6.2.5 void MVN (uint32_t ∗ *Rdn*, uint32_t *Rm*, int ∗ *flags*)

Guardar el complemento de un dato.

Parameters

Rdn	Donde se almacenara el dato
Rm	Dato al que se le hara el complemento
flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.6.2.6 void REV (uint32_t ∗ *Rdn*, uint32_t *Rm*, int ∗ *flags*)

Cambiar el orden de los bytes.

Parameters

Rdn	Donde se almacenara el resultado
Rm	Dato que se le cambiara el orden de los bytes
flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.6.2.7 void REV16 (uint32_t ∗ *Rdn*, uint32_t *Rm*, int ∗ *flags*)

Cambiar el orden de los bytes en cada halfword de 16bits.

Parameters

Rdn	Donde se almacenara el resultado
Rm	Dato que se le cambiara el orden de los bytes
flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.6.2.8 void REVSH (uint32_t ∗ *Rdn*, uint32_t *Rm*, int ∗ *flags*)

Cambiar el orden de los bytes del halfword bajo.

Parameters

∗Rdn	Donde se almacenara el resultado
Rm	Dato que se le cambiara el orden de los bytes
∗flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.6.2.9 void ROR (uint32_t ∗ *Rdn*, uint32_t *Rm*, int ∗ *flags*)

Rotacion a la derecha.

Parameters

∗Rdn	Dato que se rotara
Rm	Orden de la rotacion

Returns

No hay retorno de la funcion

5.6.2.10 void RSB (uint32_t ∗ *Rdn*, uint32_t *Rm*, int ∗ *flags*)

Complemento a dos de un dato.

Parameters

Rdn	Donde se almacenara el dato
Rm	Dato al que se le hara el complemento a dos
flags	Arreglo donde se almacenaran las banderas

Returns

No hay retorno de la funcion

5.7 Instrucciones_saltos.c File Reference

libreria encargada de realizar las funciones saltos

```
#include "Instrucciones_saltos.h"
```

Functions

- void BEQ (uint32_t ∗registro, int Salto, int ∗Banderas)

  Funcion de salto, si los datos son iguales.
- void BNE (uint32_t ∗registro, int Salto, int ∗Banderas)

  Funcion de salto, si los datos no son iguales.
- void BCS (uint32_t ∗registro, int Salto, int ∗Banderas)

  Funcion de salto, si un dato es mayor o igual a otro (sin signo)
- void BCC (uint32_t ∗registro, int Salto, int ∗Banderas)

  Funcion de salto, si un dato es menor a otro (sin signo)

- void BMI (uint32_t *registro, int Salto, int *Banderas)

  Funcion de salto, si el dato es negativo.
- void BPL (uint32_t *registro, int Salto, int *Banderas)

  Funcion de salto, si el dato es positivo.
- void BVS (uint32_t *registro, int Salto, int *Banderas)

  Funcion de salto, si hay sobreflujo.
- void BVC (uint32_t *registro, int Salto, int *Banderas)

  Funcion de salto, si no hay sobreflujo.
- void BHI (uint32_t *registro, int Salto, int *Banderas)

  Funcion de salto, si un dato es mayor que el otro (sin signo)
- void BLS (uint32_t *registro, int Salto, int *Banderas)

  Funcion de salto, si un dato es mayor o igual a otro (sin signo)
- void BGE (uint32_t *registro, int Salto, int *Banderas)

  Funcion de salto, si un dato es mayor o igual a otro (con signo)
- void BLT (uint32_t *registro, int Salto, int *Banderas)

  Funcion de salto, si un dato es menor a otro (con signo)
- void BGT (uint32_t *registro, int Salto, int *Banderas)

  Funcion de salto, si un dato es mayor a otro (con signo)
- void BLE (uint32_t *registro, int Salto, int *Banderas)

  Funcion de salto, si un dato es menor o igual a otro (con signo)
- void B (uint32_t *registro, int Salto)

  Funcion de salto, sin condicion.
- void BL (uint32_t *registro, int salto)
- void BX (uint32_t *registro, uint32_t Salto)
- void BLX (uint32_t *registro, uint32_t Registro)

5.7.1 Detailed Description

libreria encargada de realizar las funciones saltos

5.7.2 Function Documentation

5.7.2.1 void B (uint32_t * *registro,* int *Salto*)

Funcion de salto, sin condicion.

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar

Returns

No retorna

5.7.2.2 void BCC (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si un dato es menor a otro (sin signo)

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.7.2.3 void BCS (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si un dato es mayor o igual a otro (sin signo)

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.7.2.4 void BEQ (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si los datos son iguales.

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.7.2.5 void BGE (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si un dato es mayor o igual a otro (con signo)

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.7.2.6 void BGT (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si un dato es mayor a otro (con signo)

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.7.2.7 void BHI (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si un dato es mayor que el otro (sin signo)

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.7.2.8 void BL (uint32_t * *registro,* int *salto*)

5.7.2.9 void BLE (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si un dato es menor o igual a otro (con signo)

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.7.2.10 void BLS (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si un dato es mayor o igual a otro (sin signo)

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.7.2.11 void BLT (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si un dato es menor a otro (con signo)

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.7.2.12 void BLX (uint32_t * *registro,* uint32_t *Registro*)

5.7.2.13 void BMI (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si el dato es negativo.

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.7.2.14 void BNE (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si los datos no son iguales.

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.7.2.15 void BPL (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si el dato es positivo.

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.7.2.16 void BVC (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si no hay sobreflujo.

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.7.2.17 void BVS (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si hay sobreflujo.

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.7.2.18 void BX (uint32_t * *registro,* uint32_t *Salto*)

5.8 Instrucciones_saltos.h File Reference

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

Functions

- void BEQ (uint32_t *registro, int Salto, int *Banderas)
  Funcion de salto, si los datos son iguales.
- void BNE (uint32_t *registro, int Salto, int *Banderas)
  Funcion de salto, si los datos no son iguales.
- void BCS (uint32_t *registro, int Salto, int *Banderas)
  Funcion de salto, si un dato es mayor o igual a otro (sin signo)
- void BCC (uint32_t *registro, int Salto, int *Banderas)
  Funcion de salto, si un dato es menor a otro (sin signo)
- void BMI (uint32_t *registro, int Salto, int *Banderas)
  Funcion de salto, si el dato es negativo.
- void BPL (uint32_t *registro, int Salto, int *Banderas)
  Funcion de salto, si el dato es positivo.
- void BVS (uint32_t *registro, int Salto, int *Banderas)
  Funcion de salto, si hay sobreflujo.
- void BVC (uint32_t *registro, int Salto, int *Banderas)
  Funcion de salto, si no hay sobreflujo.
- void BHI (uint32_t *registro, int Salto, int *Banderas)

Funcion de salto, si un dato es mayor que el otro (sin signo)

- void BLS (uint32_t *registro, int Salto, int *Banderas)

  Funcion de salto, si un dato es mayor o igual a otro (sin signo)
- void BGE (uint32_t *registro, int Salto, int *Banderas)

  Funcion de salto, si un dato es mayor o igual a otro (con signo)
- void BLT (uint32_t *registro, int Salto, int *Banderas)

  Funcion de salto, si un dato es menor a otro (con signo)
- void BGT (uint32_t *registro, int Salto, int *Banderas)

  Funcion de salto, si un dato es mayor a otro (con signo)
- void BLE (uint32_t *registro, int Salto, int *Banderas)

  Funcion de salto, si un dato es menor o igual a otro (con signo)
- void B (uint32_t *registro, int Salto)

  Funcion de salto, sin condicion.
- void BL (uint32_t *registro, int salto)
- void BX (uint32_t *registro, uint32_t Salto)
- void BLX (uint32_t *registro, uint32_t Registro)

5.8.1 Function Documentation

5.8.1.1 void B (uint32_t * *registro,* int *Salto*)

Funcion de salto, sin condicion.

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar

Returns

No retorna

5.8.1.2 void BCC (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si un dato es menor a otro (sin signo)

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.8.1.3 void BCS (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si un dato es mayor o igual a otro (sin signo)

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.8.1.4 void BEQ (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si los datos son iguales.

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.8.1.5 void BGE (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si un dato es mayor o igual a otro (con signo)

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.8.1.6 void BGT (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si un dato es mayor a otro (con signo)

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.8.1.7 void BHI (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si un dato es mayor que el otro (sin signo)

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.8.1.8 void BL (uint32_t * *registro,* int *salto*)

5.8.1.9 void BLE (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si un dato es menor o igual a otro (con signo)

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.8.1.10 void BLS (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si un dato es mayor o igual a otro (sin signo)

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.8.1.11 void BLT (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si un dato es menor a otro (con signo)

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.8.1.12 void BLX (uint32_t * *registro,* uint32_t *Registro*)

5.8.1.13 void BMI (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si el dato es negativo.

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.8.1.14 void BNE (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si los datos no son iguales.

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.8.1.15 void BPL (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si el dato es positivo.

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.8.1.16 void BVC (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si no hay sobreflujo.

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.8.1.17 void BVS (uint32_t * *registro,* int *Salto,* int * *Banderas*)

Funcion de salto, si hay sobreflujo.

Parameters

PC	Intrucciones a la que se le hara el salto correspondiente
Saltos	correspondientes a realizar
Banderas	Quien se analizara para ver si se realiza el salto

Returns

No retorna

5.8.1.18 void BX (uint32_t * *registro,* uint32_t *Salto*)

5.9 main.c File Reference

Codigo en el cual se ejecutaran las funciones, utilizando todas las librerias creadas para cada una de ellas ademas de la libreria curses.h para la interfaz del emulador, y de la libreria decoder.h para obtener las instrucciones del documento de texto.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <curses.h>
#include "Registros.h"
#include "Instrucciones.h"
#include "Instrucciones_desplazamiento.h"
#include "decoder.h"
```

Functions

- int main (void)

5.9.1 Detailed Description

Codigo en el cual se ejecutaran las funciones, utilizando todas las librerias creadas para cada una de ellas ademas de la libreria curses.h para la interfaz del emulador, y de la libreria decoder.h para obtener las instrucciones del documento de texto.

5.9.2 Function Documentation

5.9.2.1 int main (void)

5.10 Registros.c File Reference

Contiene la funcion para poder imprimir en pantalla los registros.

```
#include "Registros.h"
#include <curses.h>
```

Functions

- void MostrarRegistro (uint32_t *registro)

  Funcion que muestra el registro.

5.10.1 Detailed Description

Contiene la funcion para poder imprimir en pantalla los registros.

5.10.2 Function Documentation

5.10.2.1 void MostrarRegistro (uint32_t * *registro*)

Funcion que muestra el registro.

Parameters

*registro	Arreglo que contiene los registros

Returns

No hay retorno de la funcion

5.11 Registros.h File Reference

Archivo que contiene la definicion de la funcion para imprimir en pantalla los registros.

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

Functions

- void MostrarRegistro (uint32_t *registro)

  Funcion que muestra el registro.

5.11.1 Detailed Description

Archivo que contiene la definicion de la funcion para imprimir en pantalla los registros.

5.11.2 Function Documentation

5.11.2.1 void MostrarRegistro (uint32_t * *registro*)

Funcion que muestra el registro.

Parameters

*registro	Arreglo que contiene los registros

Returns

No hay retorno de la funcion