Documentacion Proyecto

Generado por Doxygen 1.8.10

Martes, 20 de Octubre de 2015 07:55:33

Índice general

Capítulo 1

Índice de estructura de datos

1.1. Estructura de datos

Lista de estructuras con una breve descripción:

Capítulo 2

Indice de archivos

2.1. Lista de archivos

Lista de todos los archivos con descripciones breves:

Capítulo 3

Documentación de las estructuras de datos

3.1. Referencia de la Estructura ins_t

`#include <decoder.h>`

Campos de datos

- char ** array

3.1.1. Documentación de los campos

3.1.1.1. char** array

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

- decoder.h

3.2. 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
- uint8_t registers_list [16]

3.2.1. Documentación de los campos

3.2.1.1. char mnemonic[10]

3.2.1.2. char op1_type

3.2.1.3. uint32_t op1_value

3.2.1.4. char op2_type

3.2.1.5. uint32_t op2_value

3.2.1.6. char op3_type

3.2.1.7. uint32_t op3_value

3.2.1.8. uint8_t registers_list[16]

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

- decoder.h

3.3. Referencia de la Estructura port_t

```
#include <io.h>
```

Campos de datos

- uint8_t DDR
- uint8_t PORT
- uint8_t PIN
- uint8_t Pins
- uint8_t Interrupts

3.3.1. Documentación de los campos

3.3.1.1. uint8_t DDR

3.3.1.2. uint8_t Interrupts

3.3.1.3. uint8_t PIN

3.3.1.4. uint8_t Pins

3.3.1.5. uint8_t PORT

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

- io.h

Capítulo 4

Documentación de archivos

4.1. Referencia del Archivo alu.c

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

Funciones

- void ADD (uint32_t ∗Ra, uint32_t Rb, uint32_t Rc, bool flg[ ], int ∗pc)
  funcion que suma
- void SUB (uint32_t ∗Ra, uint32_t ∗Rb, uint32_t ∗Rc, bool ∗flg, int ∗pc)
  funcion que resta
- void MUL (uint32_t ∗Ra, uint32_t ∗Rb, uint32_t ∗Rc, bool ∗flg, int ∗pc)
  funcion que multiplica dos registros
- void AND (uint32_t ∗Ra, uint32_t ∗Rb, uint32_t ∗Rc, bool ∗flg, int ∗pc)
  funcion de producto logico
- void OR (uint32_t ∗Ra, uint32_t ∗Rb, uint32_t ∗Rc, bool ∗flg, int ∗pc)
  funcion de suma logica
- void EOR (uint32_t ∗Ra, uint32_t ∗Rb, uint32_t ∗Rc, bool ∗flg, int ∗pc)
  funcion de or exclusiva
- void MOV (uint32_t ∗Ra, uint32_t ∗Rb, bool ∗flg, int ∗pc)
  funcion que escribe un valor de un registro en otro registro

4.1.1. Documentación de las funciones

4.1.1.1. void ADD (uint32_t ∗ *Ra,* uint32_t *Rb,* uint32_t *Rc,* bool *flg[ ],* int ∗ *pc*)

funcion que suma

Parámetros

Ra	guarda el resultado
Rb	operando 1
Rc	operando 2

Devuelve

no retorna nada

4.1.1.2. void AND (uint32_t * *Ra*, uint32_t * *Rb*, uint32_t * *Rc*, bool * *flg*, int * *pc*)

funcion de producto logico

Parámetros

Ra	guarda el resultado
Rb	operando 1
Rc	operando 2

Devuelve

No retorna nada

4.1.1.3. void EOR (uint32_t * *Ra,* uint32_t * *Rb,* uint32_t * *Rc,* bool * *flg,* int * *pc*)

funcion de or exclusiva

Parámetros

Ra	guarda el resultado
Rb	operando 1
Rc	operando 2

Devuelve

No retorna nada

4.1.1.4. void MOV (uint32_t * *Ra,* uint32_t * *Rb,* bool * *flg,* int * *pc*)

funcion que escribe un valor de un registro en otro registro

Parámetros

Ra	guarda el resultado
Rb	operando 1

Devuelve

No retorna nada

4.1.1.5. void MUL (uint32_t * *Ra,* uint32_t * *Rb,* uint32_t * *Rc,* bool * *flg,* int * *pc*)

funcion que multiplica dos registros

Parámetros

Ra	guarda el resultado
Rb	operando 1
Rc	operando 2

Devuelve

No retorna nada

4.1.1.6. void OR (uint32_t * *Ra,* uint32_t * *Rb,* uint32_t * *Rc,* bool * *flg,* int * *pc*)

funcion de suma logica

Parámetros

Ra	guarda el resultado
Rb	operando 1
Rc	operando 2

Devuelve

No retorna nada

4.1.1.7. void SUB (uint32_t * *Ra,* uint32_t * *Rb,* uint32_t * *Rc,* bool * *flg,* int * *pc*)

funcion que resta

Parámetros

Ra	guaRaa el resultado
Rb	operando 1
Rc	operando 2

Devuelve

No retorna nada

4.2. Referencia del Archivo alu.h

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

Funciones

- void ADD (uint32_t *Ra, uint32_t Rb, uint32_t Rc, bool flg[ ], int *pc)
  funcion que suma
- void SUB (uint32_t *Ra, uint32_t *Rb, uint32_t *Rc, bool *flg, int *pc)
  funcion que resta
- void AND (uint32_t *Ra, uint32_t *Rb, uint32_t *Rc, bool *flg, int *pc)
  funcion de producto logico
- void OR (uint32_t *Ra, uint32_t *Rb, uint32_t *Rc, bool *flg, int *pc)
  funcion de suma logica
- void EOR (uint32_t *Ra, uint32_t *Rb, uint32_t *Rc, bool *flg, int *pc)
  funcion de or exclusiva
- void MOV (uint32_t *Ra, uint32_t *Rb, bool *flg, int *pc)
  funcion que escribe un valor de un registro en otro registro
- void MUL (uint32_t *Ra, uint32_t *Rb, uint32_t *Rc, bool *flg, int *pc)
  funcion que multiplica dos registros

4.2.1. Documentación de las funciones

4.2.1.1. void ADD (uint32_t * *Ra,* uint32_t *Rb,* uint32_t *Rc,* bool *flg[ ],* int * *pc*)

funcion que suma

Parámetros

Ra	guarda el resultado
Rb	operando 1
Rc	operando 2

Devuelve

no retorna nada

4.2.1.2. void AND (uint32_t * *Ra*, uint32_t * *Rb*, uint32_t * *Rc*, bool * *flg*, int * *pc*)

funcion de producto logico

Parámetros

Ra	guarda el resultado
Rb	operando 1
Rc	operando 2

Devuelve

No retorna nada

4.2.1.3. void EOR (uint32_t * *Ra*, uint32_t * *Rb*, uint32_t * *Rc*, bool * *flg*, int * *pc*)

funcion de or exclusiva

Parámetros

Ra	guarda el resultado
Rb	operando 1
Rc	operando 2

Devuelve

No retorna nada

4.2.1.4. void MOV (uint32_t * *Ra*, uint32_t * *Rb*, bool * *flg*, int * *pc*)

funcion que escribe un valor de un registro en otro registro

Parámetros

Ra	guarda el resultado
Rb	operando 1

Devuelve

No retorna nada

4.2.1.5. void MUL (uint32_t * *Ra*, uint32_t * *Rb*, uint32_t * *Rc*, bool * *flg*, int * *pc*)

funcion que multiplica dos registros

Parámetros

Ra	guarda el resultado
Rb	operando 1
Rc	operando 2

Devuelve

No retorna nada

4.2.1.6. void OR (uint32_t * *Ra,* uint32_t * *Rb,* uint32_t * *Rc,* bool * *flg,* int * *pc*)

funcion de suma logica

Parámetros

Ra	guarda el resultado
Rb	operando 1
Rc	operando 2

Devuelve

No retorna nada

4.2.1.7. void SUB (uint32_t * *Ra,* uint32_t * *Rb,* uint32_t * *Rc,* bool * *flg,* int * *pc*)

funcion que resta

Parámetros

Ra	guaRaa el resultado
Rb	operando 1
Rc	operando 2

Devuelve

No retorna nada

4.3. Referencia del Archivo banderas.c

```
#include "banderas.h"
```

Funciones

- void banderas (uint32_t Ra, uint32_t Rb, uint32_t Rc, bool flg[ ])
  funcion que registran los estados del microprocesador

4.3.1. Documentación de las funciones

4.3.1.1. void banderas (uint32_t *Ra,* uint32_t *Rb,* uint32_t *Rc,* bool *flg[ ]*)

funcion que registran los estados del microprocesador

Parámetros

Ra	operando 1
Rb	operando 2
Rc	operando 3

Devuelve

no retorna nada

4.4. Referencia del Archivo banderas.h

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

Funciones

- void banderas (uint32_t Ra, uint32_t Rb, uint32_t Rc, bool flg[ ])
  funcion que registran los estados del microprocesador

4.4.1. Documentación de las funciones

4.4.1.1. void banderas (uint32_t *Ra,* uint32_t *Rb,* uint32_t *Rc,* bool *flg[ ]*)

funcion que registran los estados del microprocesador

Parámetros

Ra	operando 1
Rb	operando 2
Rc	operando 3

Devuelve

no retorna nada

4.5. Referencia del Archivo decoder.c

```
#include "decoder.h"
#include "alu.h"
#include "banderas.h"
#include "InstruccionesDesplazamiento.h"
#include <string.h>
#include "RAM.h"
```

Funciones

- void decodeInstruction (instruction_t instruction, uint32_t ∗Rd, uint32_t ∗Rm, uint32_t ∗Rr, bool flg[ ], int ∗pc, uint8_t pila)

- instruction_t getInstruction (char *instStr)
- int readFile (char *filename, ins_t *instructions)
- int countLines (FILE *fp)

4.5.1. Documentación de las funciones

4.5.1.1. int countLines (FILE * *fp*)

4.5.1.2. void decodeInstruction (instruction_t *instruction,* uint32_t * *Rd,* uint32_t * *Rm,* uint32_t * *Rr,* bool *flg[ ],* int * *pc,* uint8_t *pila*)

4.5.1.3. instruction_t getInstruction (char * *instStr*)

4.5.1.4. int readFile (char * *filename,* ins_t * *instructions*)

4.6. Referencia del Archivo decoder.h

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

Estructuras de datos

- struct ins_t
- struct instruction_t

Funciones

- int readFile (char *filename, ins_t *instructions)
- int countLines (FILE *fp)

4.6.1. Documentación de las funciones

4.6.1.1. int countLines (FILE * *fp*)

4.6.1.2. int readFile (char * *filename,* ins_t * *instructions*)

4.7. Referencia del Archivo InstruccionesDesplazamiento.c

```
#include "InstruccionesDesplazamiento.h"
```

Funciones

- void LSL (uint32_t *Ra, uint32_t *Rb, uint32_t inmediato, int *pc)

  funcion de desplazamiento logico a la izquierda
- void NOP (int *pc)

  funcion que no hace ninguna operacion solo aumenta el valor del pc

- void LSR (uint32_t ∗Ra, uint32_t ∗Rb, uint32_t inmediato, int ∗pc)

  funcion de desplazamiento logico a la derecha
- void BIC (uint32_t ∗Ra, uint32_t ∗Rb, int ∗pc)

  funcion que realiza una AND entre un registro y el complemento de otro
- void MVN (uint32_t ∗Ra, uint32_t ∗Rb, int ∗pc)

  funcion que gua∗Raa el complemento de un numero
- void RSBS (uint32_t ∗Ra, uint32_t ∗Rb, int ∗pc)

  funcion que obtiene el complemento a dos de un numero
- void ASRS (uint32_t ∗Ra, uint32_t ∗Rb, int ∗pc)

  funcion de desplazamiento aritmetico a la derecha
- void ROR (uint32_t ∗Ra, uint32_t ∗Rb, int ∗pc)

  funcion de rotacion a la derecha
- void CMP (uint32_t ∗Ra, uint32_t ∗Rb, uint32_t ∗Rc, int ∗pc)

  funcion de resta que solo modifica las banderas
- void CMN (uint32_t ∗Ra, uint32_t ∗Rb, uint32_t ∗Rc, int ∗pc)

  funcion de suma que solo modifica las banderas

4.7.1. Documentación de las funciones

4.7.1.1. void ASRS (uint32_t ∗ *Ra*, uint32_t ∗ *Rb*, int ∗ *pc*)

funcion de desplazamiento aritmetico a la derecha

Parámetros

Ra	operando 1
Rb	operando 2

Devuelve

no retorna nada

4.7.1.2. void BIC (uint32_t ∗ *Ra*, uint32_t ∗ *Rb*, int ∗ *pc*)

funcion que realiza una AND entre un registro y el complemento de otro

Parámetros

Ra	operando 1
Rb	operando 2

Devuelve

no retorna nada

4.7.1.3. void CMN (uint32_t ∗ *Ra*, uint32_t ∗ *Rb*, uint32_t ∗ *Rc*, int ∗ *pc*)

funcion de suma que solo modifica las banderas

Parámetros

Ra	operando 1
Rb	operando 2
Rc	operando 3
pc	operando 4

Devuelve

no retorna nada

4.7.1.4. void CMP (uint32_t * *Ra,* uint32_t * *Rb,* uint32_t * *Rc,* int * *pc*)

funcion de resta que solo modifica las banderas

Parámetros

Ra	operando 1
Rb	operando 2
Rc	operando 3
pc	operando 4

Devuelve

no retorna nada

4.7.1.5. void LSL (uint32_t * *Ra,* uint32_t * *Rb,* uint32_t *inmediato,* int * *pc*)

funcion de desplazamiento logico a la izquierda

Parámetros

Ra	operando 1, ademas guarda el resultado
Rb	operando 2
inmediato	operando 3

Devuelve

no retorna nada

4.7.1.6. void LSR (uint32_t * *Ra,* uint32_t * *Rb,* uint32_t *inmediato,* int * *pc*)

funcion de desplazamiento logico a la derecha

Parámetros

Ra	operando1, ademas guarda el resultado
Rb	operando 2
inmediato	operando 3

Devuelve

no retorna nada

4.7.1.7. void MVN (uint32_t * *Ra,* uint32_t * *Rb,* int * *pc*)

funcion que gua*Raa el complemento de un numero

Parámetros

Ra	operando 1
Rb	operando 2

Devuelve

no retorna nada

4.7.1.8. void NOP (int * *pc*)

funcion que no hace ninguna operacion solo aumenta el valor del pc

4.7.1.9. void ROR (uint32_t * *Ra,* uint32_t * *Rb,* int * *pc*)

funcion de rotacion a la derecha

Parámetros

Ra	operando 1
Rb	operando 2

Devuelve

no retorna nada

4.7.1.10. void RSBS (uint32_t * *Ra,* uint32_t * *Rb,* int * *pc*)

funcion que obtiene el complemento a dos de un numero

Parámetros

Ra	operando 1
Rb	operando 2

Devuelve

no retorna nada

4.8. Referencia del Archivo InstruccionesDesplazamiento.h

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

Funciones

- void LSL (uint32_t *Ra, uint32_t *Rb, uint32_t inmediato, int *pc)

  funcion de desplazamiento logico a la izquierda
- void LSR (uint32_t *Ra, uint32_t *Rb, uint32_t inmediato, int *pc)

  funcion de desplazamiento logico a la derecha
- void BIC (uint32_t *Ra, uint32_t *Rb, int *pc)

funcion que realiza una AND entre un registro y el complemento de otro

- void MVN (uint32_t ∗Ra, uint32_t ∗Rb, int ∗pc)

  funcion que gua∗Raa el complemento de un numero
- void RSBS (uint32_t ∗Ra, uint32_t ∗Rb, int ∗pc)

  funcion que obtiene el complemento a dos de un numero
- void NOP (int ∗pc)

  funcion que no hace ninguna operacion solo aumenta el valor del pc
- void ASRS (uint32_t ∗Ra, uint32_t ∗Rb, int ∗pc)

  funcion de desplazamiento aritmetico a la derecha
- void ROR (uint32_t ∗Ra, uint32_t ∗Rb, int ∗pc)

  funcion de rotacion a la derecha
- void CMP (uint32_t ∗Ra, uint32_t ∗Rb, uint32_t ∗Rc, int ∗pc)

  funcion de resta que solo modifica las banderas
- void CMN (uint32_t ∗Ra, uint32_t ∗Rb, uint32_t ∗Rc, int ∗pc)

  funcion de suma que solo modifica las banderas

4.8.1. Documentación de las funciones

4.8.1.1. void ASRS (uint32_t ∗ *Ra,* uint32_t ∗ *Rb,* int ∗ *pc*)

funcion de desplazamiento aritmetico a la derecha

Parámetros

Ra	operando 1
Rb	operando 2

Devuelve

no retorna nada

4.8.1.2. void BIC (uint32_t ∗ *Ra,* uint32_t ∗ *Rb,* int ∗ *pc*)

funcion que realiza una AND entre un registro y el complemento de otro

Parámetros

Ra	operando 1
Rb	operando 2

Devuelve

no retorna nada

4.8.1.3. void CMN (uint32_t ∗ *Ra,* uint32_t ∗ *Rb,* uint32_t ∗ *Rc,* int ∗ *pc*)

funcion de suma que solo modifica las banderas

Parámetros

Ra	operando 1
Rb	operando 2
Rc	operando 3
pc	operando 4

Devuelve

no retorna nada

4.8.1.4. void CMP (uint32_t * *Ra,* uint32_t * *Rb,* uint32_t * *Rc,* int * *pc*)

funcion de resta que solo modifica las banderas

Parámetros

Ra	operando 1
Rb	operando 2
Rc	operando 3
pc	operando 4

Devuelve

no retorna nada

4.8.1.5. void LSL (uint32_t * *Ra,* uint32_t * *Rb,* uint32_t *inmediato,* int * *pc*)

funcion de desplazamiento logico a la izquierda

Parámetros

Ra	operando 1, ademas guarda el resultado
Rb	operando 2
inmediato	operando 3

Devuelve

no retorna nada

4.8.1.6. void LSR (uint32_t * *Ra,* uint32_t * *Rb,* uint32_t *inmediato,* int * *pc*)

funcion de desplazamiento logico a la derecha

Parámetros

Ra	operando1, ademas guarda el resultado
Rb	operando 2
inmediato	operando 3

Devuelve

no retorna nada

4.8.1.7. void MVN (uint32_t * *Ra,* uint32_t * *Rb,* int * *pc*)

funcion que gua*Raa el complemento de un numero

Parámetros

Ra	operando 1
Rb	operando 2

Devuelve

no retorna nada

4.8.1.8. void NOP (int * *pc*)

funcion que no hace ninguna operacion solo aumenta el valor del pc

4.8.1.9. void ROR (uint32_t * *Ra,* uint32_t * *Rb,* int * *pc*)

funcion de rotacion a la derecha

Parámetros

Ra	operando 1
Rb	operando 2

Devuelve

no retorna nada

4.8.1.10. void RSBS (uint32_t * *Ra,* uint32_t * *Rb,* int * *pc*)

funcion que obtiene el complemento a dos de un numero

Parámetros

Ra	operando 1
Rb	operando 2

Devuelve

no retorna nada

4.9. Referencia del Archivo interrupciones.c

4.10. Referencia del Archivo interrupciones.h

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

4.11. Referencia del Archivo io.c

```
#include "io.h"
```

Funciones

- void initIO (void)
- void changePinPortA (uint8_t pin, uint8_t value)
- void changePinPortB (uint8_t pin, uint8_t value)
- void IOAccess (uint8_t address, uint8_t *data, uint8_t r_w)
- void showPorts (void)
- void showFrame (int x, int y, int w, int h)

Variables

- port_t PORTA
- port_t PORTB
- uint8_t irq [16]

4.11.1. Documentación de las funciones

4.11.1.1. void changePinPortA (uint8_t *pin,* uint8_t *value*)

4.11.1.2. void changePinPortB (uint8_t *pin,* uint8_t *value*)

4.11.1.3. void initIO (void)

4.11.1.4. void IOAccess (uint8_t *address,* uint8_t * *data,* uint8_t *r_w*)

4.11.1.5. void showFrame (int *x,* int *y,* int *w,* int *h*)

4.11.1.6. void showPorts (void)

4.11.2. Documentación de las variables

4.11.2.1. uint8_t irq[16]

4.11.2.2. port_t PORTA

4.11.2.3. port_t PORTB

4.12. Referencia del Archivo io.h

```
#include <stdint.h>
#include <curses.h>
```

Estructuras de datos

- struct port_t

'defines'

- #define XINIT 10
- #define YINIT 5
- #define HIGH 1
- #define LOW 0

- #define Read 1
- #define Write 0
- #define BLUEBLACK 10 /∗Text Blue Background Black∗/
- #define REDBLACK 20 /∗Text Red Background Black∗/
- #define WHITEBLACK 30 /∗Text White Background White∗/

Funciones

- void IOAccess (uint8_t address, uint8_t ∗data, uint8_t r_w)
- void changePinPortA (uint8_t pin, uint8_t value)
- void changePinPortB (uint8_t pin, uint8_t value)
- void initIO (void)
- void showPorts (void)
- void showFrame (int x, int y, int w, int h)

4.12.1. Documentación de los 'defines'

4.12.1.1. #define BLUEBLACK 10 /∗Text Blue Background Black∗/

4.12.1.2. #define HIGH 1

4.12.1.3. #define LOW 0

4.12.1.4. #define Read 1

4.12.1.5. #define REDBLACK 20 /∗Text Red Background Black∗/

4.12.1.6. #define WHITEBLACK 30 /∗Text White Background White∗/

4.12.1.7. #define Write 0

4.12.1.8. #define XINIT 10

4.12.1.9. #define YINIT 5

4.12.2. Documentación de las funciones

4.12.2.1. void changePinPortA (uint8_t *pin,* uint8_t *value*)

4.12.2.2. void changePinPortB (uint8_t *pin,* uint8_t *value*)

4.12.2.3. void initIO (void)

4.12.2.4. void IOAccess (uint8_t *address,* uint8_t ∗ *data,* uint8_t *r_w*)

4.12.2.5. void showFrame (int *x,* int *y,* int *w,* int *h*)

4.12.2.6. void showPorts (void)

4.13. Referencia del Archivo Load_Store.c

```
#include "Load_Store.h"
#include "ram.h"
```

Funciones

- void LDR (uint32_t *Rt, uint32_t Rn, uint32_t inmed, int *pc)
- void LDRB (uint32_t *Rt, uint32_t Rn, uint32_t inmed, int *pc)
- void LDRH (uint32_t *Rt, uint32_t Rn, uint32_t inmed, int *pc)
- void LDRSB (uint32_t *Rt, uint32_t Rn, uint32_t inmed, int *pc)
- void LDRSH (uint32_t *Rt, uint32_t Rn, uint32_t inmed, int *pc)
- void STR (uint32_t *Rt, uint32_t Rn, uint32_t inmed, uint32_t SP, int *pc)
- void STRB (uint32_t *Rt, uint32_t Rn, uint32_t inmed, uint32_t SP, int *pc)
- void STRH (uint32_t *Rt, uint32_t Rn, uint32_t inmed, uint32_t SP, int *pc)

4.13.1. Documentación de las funciones

4.13.1.1. void LDR (uint32_t * *Rt*, uint32_t *Rn*, uint32_t *inmed*, int * *pc*)

4.13.1.2. void LDRB (uint32_t * *Rt*, uint32_t *Rn*, uint32_t *inmed*, int * *pc*)

4.13.1.3. void LDRH (uint32_t * *Rt*, uint32_t *Rn*, uint32_t *inmed*, int * *pc*)

4.13.1.4. void LDRSB (uint32_t * *Rt*, uint32_t *Rn*, uint32_t *inmed*, int * *pc*)

4.13.1.5. void LDRSH (uint32_t * *Rt*, uint32_t *Rn*, uint32_t *inmed*, int * *pc*)

4.13.1.6. void STR (uint32_t * *Rt*, uint32_t *Rn*, uint32_t *inmed*, uint32_t *SP*, int * *pc*)

4.13.1.7. void STRB (uint32_t * *Rt*, uint32_t *Rn*, uint32_t *inmed*, uint32_t *SP*, int * *pc*)

4.13.1.8. void STRH (uint32_t * *Rt*, uint32_t *Rn*, uint32_t *inmed*, uint32_t *SP*, int * *pc*)

4.14. Referencia del Archivo Load_Store.h

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

Funciones

- void LDR (uint32_t Rt, uint32_t Rn, uint32_t inmed)

  funcion de Carga
- void LDRB (uint32_t Rt, uint32_t Rn, uint32_t inmed)

  funcion de Carga con extension de cero
- void LDRH (uint32_t Rt, uint32_t Rn, uint32_t inmed)

  funcion de Carga con extension de cero
- void LDRSB (uint32_t Rt, uint32_t Rn, uint32_t inmed)

  funcion de Carga con extension de signo
- void LDRSH (uint32_t Rt, uint32_t Rn, uint32_t inmed)

  funcion de Carga con extension de signo
- void STR (uint32_t Rt, uint32_t Rn, uint32_t inmed, uint32_t SP)

  funcion de almacenamiento
- void STRB (uint32_t Rt, uint32_t Rn, uint32_t inmed, uint32_t SP)

  funcion de almacenamiento
- void STRH (uint32_t Rt, uint32_t Rn, uint32_t inmed, uint32_t SP)

  funcion de almacenamiento

4.14.1. Documentación de las funciones

4.14.1.1. void LDR (uint32_t *Rt,* uint32_t *Rn,* uint32_t *inmed*)

funcion de Carga

Parámetros

Rt	guarda el resultado
Rn	operando 1
inmed	operando 2

Devuelve

no retorna nada

4.14.1.2. void LDRB (uint32_t *Rt,* uint32_t *Rn,* uint32_t *inmed*)

funcion de Carga con extension de cero

Parámetros

Rt	guarda el resultado
Rn	operando 1
inmed	operando 2

Devuelve

no retorna nada

4.14.1.3. void LDRH (uint32_t *Rt,* uint32_t *Rn,* uint32_t *inmed*)

funcion de Carga con extension de cero

Parámetros

Rt	guarda el resultado
Rn	operando 1
inmed	operando 2

Devuelve

no retorna nada

4.14.1.4. void LDRSB (uint32_t *Rt,* uint32_t *Rn,* uint32_t *inmed*)

funcion de Carga con extension de signo

Parámetros

Rt	guarda el resultado
Rn	operando 1

inmed	operando 2

Devuelve

no retorna nada

4.14.1.5. void LDRSH (uint32_t *Rt,* uint32_t *Rn,* uint32_t *inmed*)

funcion de Carga con extension de signo

Parámetros

Rt	guarda el resultado
Rn	operando 1
inmed	operando 2

Devuelve

no retorna nada

4.14.1.6. void STR (uint32_t *Rt,* uint32_t *Rn,* uint32_t *inmed,* uint32_t *SP*)

funcion de almacenamiento

Parámetros

Rt	operando 1
Rn	operando 2
inmed	operando 3
SP	operando 4

Devuelve

no retorna nada

4.14.1.7. void STRB (uint32_t *Rt,* uint32_t *Rn,* uint32_t *inmed,* uint32_t *SP*)

funcion de almacenamiento

Parámetros

Rt	operando 1
Rn	operando 2
inmed	operando 3
SP	operando 4

Devuelve

no retorna nada

4.14.1.8. void STRH (uint32_t *Rt,* uint32_t *Rn,* uint32_t *inmed,* uint32_t *SP*)

funcion de almacenamiento

Parámetros

Rt	operando 1
Rn	operando 2
inmed	operando 3
SP	operando 4

Devuelve

no retorna nada

4.15. Referencia del Archivo main.c

```
#include "registros.h"
#include "decoder.h"
#include <curses.h>
```

Funciones

- int main (void)

4.15.1. Documentación de las funciones

4.15.1.1. int main (void)

4.16. Referencia del Archivo ports.c

```
#include "io.h"
```

Funciones

- int main (void)

Variables

- uint8_t irq [16]

4.16.1. Documentación de las funciones

4.16.1.1. int main (void)

4.16.2. Documentación de las variables

4.16.2.1. uint8_t irq[16]

4.17. Referencia del Archivo RAM.c

```
#include <stdint.h>
#include "RAM.h"
#include <stdio.h>
```

Funciones

- uint8_t bitcount (uint8_t listaregistros[ ])
- void push (uint8_t ∗pila, uint32_t ∗R, uint8_t ∗ram, uint8_t listaregistros[ ])
- void pop (uint8_t ∗pila, uint32_t ∗R, uint8_t ∗ram, uint8_t listaregistros[ ])

4.17.1. Documentación de las funciones

4.17.1.1. uint8_t bitcount (uint8_t *listaregistros[ ]*)

4.17.1.2. void pop (uint8_t ∗ *pila,* uint32_t ∗ *R,* uint8_t ∗ *ram,* uint8_t *listaregistros[ ]*)

4.17.1.3. void push (uint8_t ∗ *pila,* uint32_t ∗ *R,* uint8_t ∗ *ram,* uint8_t *listaregistros[ ]*)

4.18. Referencia del Archivo RAM.h

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

'defines'

- #define Mema 64

Funciones

- void push (uint8_t ∗pila, uint32_t ∗R, uint8_t ∗ram, uint8_t listaregistros[ ])
- void pop (uint8_t ∗pila, uint32_t ∗R, uint8_t ∗ram, uint8_t listaregistros[ ])
- uint8_t bitcount (uint8_t listaregistros[ ])

4.18.1. Documentación de los 'defines'

4.18.1.1. #define Mema 64

4.18.2. Documentación de las funciones

4.18.2.1. uint8_t bitcount (uint8_t *listaregistros[ ]*)

4.18.2.2. void pop (uint8_t ∗ *pila,* uint32_t ∗ *R,* uint8_t ∗ *ram,* uint8_t *listaregistros[ ]*)

4.18.2.3. void push (uint8_t ∗ *pila,* uint32_t ∗ *R,* uint8_t ∗ *ram,* uint8_t *listaregistros[ ]*)

4.19. Referencia del Archivo registros.c

```
#include "registros.h"
```

Funciones

- void Registros (uint32_t *Ra, uint32_t *Rb, uint32_t *Rc)
  funcion que muestra los registros

4.19.1. Documentación de las funciones

4.19.1.1. void Registros (uint32_t * *Ra,* uint32_t * *Rb,* uint32_t * *Rc*)

funcion que muestra los registros

Parámetros

Rd	operando 1
Rm	operando 2
Rr	operando 3

Devuelve

no retorna nada

4.20. Referencia del Archivo registros.h

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

Funciones

- void Registros (uint32_t *Ra, uint32_t *Rb, uint32_t *Rc)
  funcion que muestra los registros

4.20.1. Documentación de las funciones

4.20.1.1. void Registros (uint32_t * *Ra,* uint32_t * *Rb,* uint32_t * *Rc*)

funcion que muestra los registros

Parámetros

Rd	operando 1
Rm	operando 2
Rr	operando 3

Devuelve

no retorna nada

4.21. Referencia del Archivo saltos.c

```
#include "banderas.h"
#include "alu.h"
#include <curses.h>
```

Funciones

- void B (uint32_t valor, int *pc, bool *flg)

Variables

- uint32_t AUX

4.21.1. Documentación de las funciones

4.21.1.1. void B (uint32_t *valor,* int * *pc,* bool * *flg*)

4.21.2. Documentación de las variables

4.21.2.1. uint32_t AUX

4.22. Referencia del Archivo saltos.h

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

Funciones

- void B (uint32_t valor, int *pc, bool *flg)
- void BEQ (uint32_t valor, int *pc, bool *flg)
- void BNE (uint32_t valor, int *pc, bool *flg)
- void BCS (uint32_t valor, int *pc, bool *flg)
- void BCC (uint32_t valor, int *pc, bool *flg)
- void BMI (uint32_t valor, int *pc, bool *flg)
- void BPL (uint32_t valor, int *pc, bool *flg)
- void BVS (uint32_t valor, int *pc, bool *flg)
- void BVC (uint32_t valor, int *pc, bool *flg)
- void BHI (uint32_t valor, int *pc, bool *flg)
- void BLS (uint32_t valor, int *pc, bool *flg)
- void BGE (uint32_t valor, int *pc, bool *flg)
- void BLT (uint32_t valor, int *pc, bool *flg)
- void BGT (uint32_t valor, int *pc, bool *flg)
- void BLE (uint32_t valor, int *pc, bool *flg)
- void BAL (uint32_t valor, int *pc, bool *flg)
- void BL (uint32_t valor, int *pc, bool *flg)
- void BX (uint32_t *pc, bool *flg)

4.22.1. Documentación de las funciones

4.22.1.1. void B (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.2. void BAL (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.3. void BCC (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.4. void BCS (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.5. void BEQ (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.6. void BGE (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.7. void BGT (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.8. void BHI (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.9. void BL (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.10. void BLE (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.11. void BLS (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.12. void BLT (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.13. void BMI (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.14. void BNE (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.15. void BPL (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.16. void BVC (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.17. void BVS (uint32_t *valor,* int * *pc,* bool * *flg*)

4.22.1.18. void BX (uint32_t * *pc,* bool * *flg*)

4.23. Referencia del Archivo test.c