

# Register File especificación DCILAB

Edgar Solera Bolaños

Agosto 2018

## 1. Tabla

Cuadro 1: CSR registers Directions.

Register	Implementation Dir.
Registro configuración	0x00
mepc	0x01
Interrup1	0x02
Interrupt2	0x03
GPIO	0x04
mvtec	0x05
comparación timer	0x06
Valor Timer	0x07
Full Range Level Shifter	0x08
IS_Valr	0x09
IS_Config	0x10
IS_Trigger	0x11

Los colores azules significan bit en tierra y lo rojos significan bits en VDD.

## 2. MIE/MIP

- **MIET:** time sirve para encender y apagar el timer.
- **MIPT:** se escribe solo por hardware, y se activa siempre que el timer alla superado el valor de comparación.
- **MIEIO:** habilita al sistema a atender las interrupciones de UART o SPI.
- **MIPIO:** se escribe con uno al ocurrir una interrupción de UART o SPI.
- **MAIE:** habilita las interrupciones del bloque analógico.
- **MAIP:** se escribe por hardware con un 1 cuando ocurre una interrupción del bloque analógico.

- **GPIO enable bits:** son 8 bits que cada uno controla el modo de operación de un registro GPIO. Al estar en 1 habilita la escritura desde el puerto externo, al ser 0 habilita al puerto GPIO al modo lectura.
- **bus\_en:** este bit se encarga de habilitar y deshabilitar el reloj del bus. Al ser un 1 habilita el reloj y al ser un 0 lo apaga.



Figura 1: Constitución Registro dirección 0

### 3. mepc

Al ocurrir un interrupción en el sistema , el registro mepc es escrito con el valor de la dirección del pc donde encontró dicha interrupción. Este registro tiene los 2 primeros bits cableado a 0, ya que almacena solo numero múltiplos de 4.

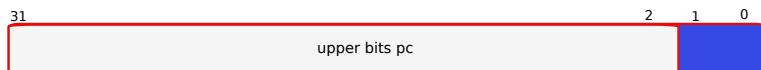


Figura 2: Constitución Registro dirección 1

### 4. interrupt1 y interrupt2

El código de las interrupciones es de 62 bits y es almacenado en 2 registros CSR. El interrupt1 contiene el Dato que es 32 bits. El interrupt2 en sus primeros 24 bits tiene la dirección, en sus próximos 3 bits tiene el código, los 2 siguientes bits contienen la fuente(source) y los restantes sobran por lo cual son cableado a 0.

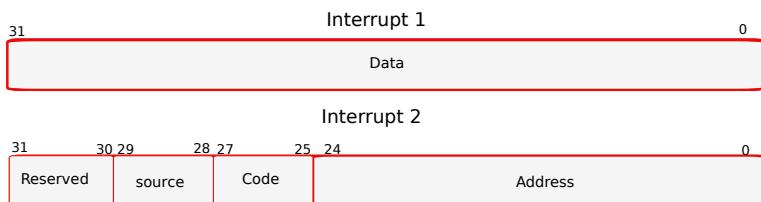


Figura 3: Registros de Interrupción

### 5. GPIO

En este espacio se encuentra los registro GPIO los cuales tienen dos modos de operación: escritura desde el puerto y lectura desde el puerto. En escritura desde el puerto el registro se escribe desde el pad del chip y en lectura desde el puerto el pad sirve para leer el valor del registro. El modo de operación es determinado por los 8 bits GPIO conf del registro de configuración. El tamaño de este registro es de 8 bits como se observa en la imagen 4 y su arquitectura de operación se observa en la imagen 5.



Figura 4: Registro de GPIO

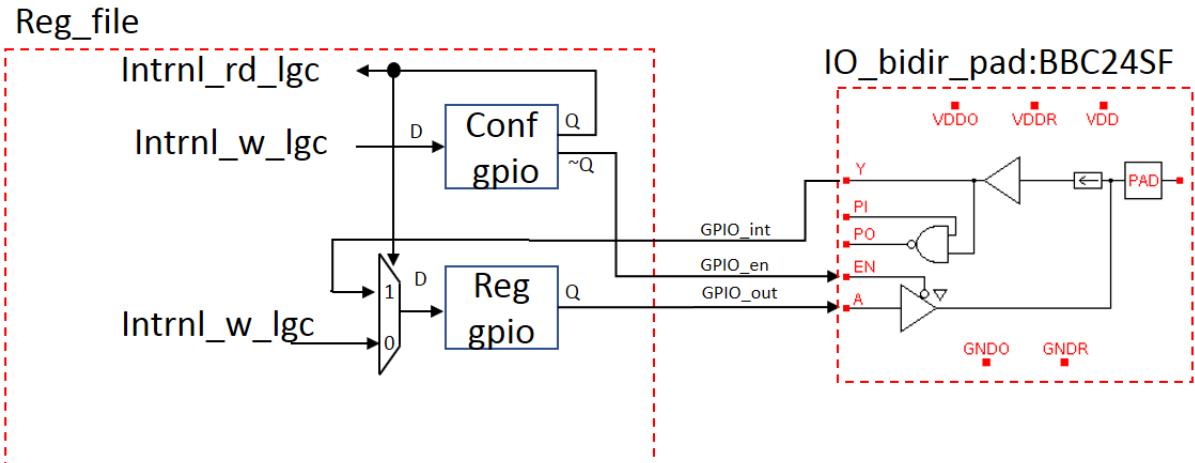


Figura 5: Aquitectura de funcionamiento

## 6. mvtec

Este registro contiene la dirección en memoria en donde el procesador debe brincar cuando un trap a ocurrido. Los primeros 2 bits no forman parte de la dirección, como se explicaba en el registro mepc, el pc solo se mueve en múltiplos de 4. Estos dos primeros bits son usado para setear el modo de operación de este registro, se cablean a 0 para setear el modo de operación estático en donde todas las interrupciones que llegan provocaran que el pc salte a la misma dirección dada por el resto de bits del mvtec.

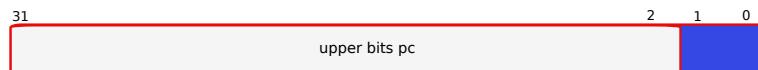


Figura 6: Constitución Registro dirección 5

## 7. Valor timer y registro comparación

Este Registro almacena del valor de timer que esta mapeado en memoria. Mientras que el registro comparación contiene el valor al que el timer debe llegar para activar la interrupción.

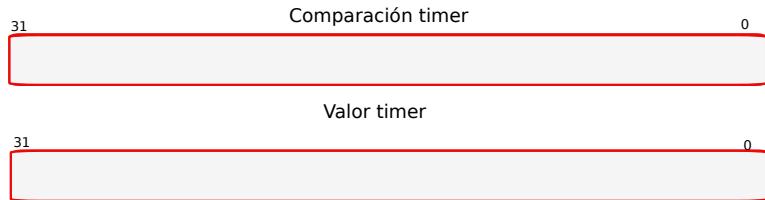


Figura 7: Constitución Registro dirección 6 y 7

## 8. Full Range Level Shift

Es un registro con 8 bits escribibles y leíble por software, que además tiene una salida secundaria directa al IO analógico del sistema. El resto de bits son un 0 lógico cableado.



Figura 8: Arquitectura registro Full Range Level Shift

## 9. Registro IS\_Var

Es un registro de 32 bits escribible y leíble por software, que además tiene una salida secundaria directa al IO analógico del sistema.



Figura 9: Registro IS\_Var

## 10. Registro IS\_Config

Es un registro de 32 bits escribible y leíble por software, que además tiene una salida secundaria directa al IO analógico del sistema.



Figura 10: Registro IS\_Config

## 11. Registro IS\_Trigger

Es un registro con 4 bits escribibles y leíble por software, que además tiene una salida secundaria directa al IO analógico del sistema. El resto de bits son un 0 lógico cableado.



Figura 11: Registro IS\_Trigger