|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table of Content** | | | | |
| **Table of Content** | | | | |
| **Index** | **Component** | **Default** | **Address** | **Page** |
| 1 | [stopwatch](#bookmark1) |  | 0x0000000 - 0x00003FF | 1 |
| 1.1 | [value](#bookmark2) | 0x0 | 0x0000000 | 1 |
| 1.2 | [reset\_value](#bookmark3) | 0x0 | 0x0000004 | 1 |
| 1.3 | [upper\_limit](#bookmark4) | 0x0 | 0x0000008 | 1 |
| 1.4 | [lower\_limit](#bookmark5) | 0x0 | 0x000000C | 1 |
| 1.5 | [csr](#bookmark6) | 0x0 | 0x0000010 | 2 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 stopwatch | | stopwatch | | block | 0x0000000 |
| offset |  | external |  | size | 0x400 |
|  | | | | | |
| block[@name='stopwatch'] | | | | | |
| **{coverage=on}** | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.1 value | | | | | | | | | | | | | | | | value | | | | | | | | | | | | | reg32 | | | | | | 0x0000000 | | | | | |
| offset | | | | | | |  | | | | | | | | | external | | | | | | |  | | | | | | default | | | | | | 0x0 | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| block[@name='stopwatch']/reg[@name='value'] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| This register contains value that can be read from the external bus during reset. There is a check on the value written in the register which can't exceed upper\_limit register value and lower than lower\_limit register. This register have only read permission during normal operation. It has one 32 bit field. **{hdl\_path= value\_rd\_data}** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | | 29 | 28 | 27 | 26 | | | 25 | 24 | | 23 | 22 | 21 | | | 20 | 19 | 18 | | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | | 9 | 8 | 7 | 6 | 5 | | 4 | 3 | 2 | 1 | 0 |
| bits | | name | | | | | | s/w | | | h/w | | | | default | | | | | description | | | | | | | | | | | | | | | | | | | | |
| 31:0 | | Field | | | | | | rw | | | ro | | | | 0x0 | | | | |  | | | | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.2 reset\_value | | | | | | | | | | | | | | | | reset\_value | | | | | | | | | | | | | reg32 | | | | | | 0x0000004 | | | | | |
| offset | | | | | | |  | | | | | | | | | external | | | | | | |  | | | | | | default | | | | | | 0x0 | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| block[@name='stopwatch']/reg[@name='reset\_value'] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| This register contains reset value that is written into value register during reset. Under normal condition it has read/write permission. **{hdl\_path=reset\_value\_rd\_data}** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | | 29 | 28 | 27 | 26 | | | 25 | 24 | | 23 | 22 | 21 | | | 20 | 19 | 18 | | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | | 9 | 8 | 7 | 6 | 5 | | 4 | 3 | 2 | 1 | 0 |
| bits | | name | | | | | | s/w | | | h/w | | | | default | | | | | description | | | | | | | | | | | | | | | | | | | | |
| 31:0 | | field | | | | | | rw | | | wo | | | | 0x0 | | | | | **{hdl\_path= reset\_value\_field\_q}** | | | | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.3 upper\_limit | | | | | | | | | | | | | | | | upper\_limit | | | | | | | | | | | | | reg32 | | | | | | 0x0000008 | | | | | |
| offset | | | | | | |  | | | | | | | | | external | | | | | | |  | | | | | | default | | | | | | 0x0 | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| block[@name='stopwatch']/reg[@name='upper\_limit'] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| The value written in this register is used to check the value in the value register which can't be greater than upper\_limit value. It has r/w permission under normal condition. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | | 29 | 28 | 27 | 26 | | | 25 | 24 | | 23 | 22 | 21 | | | 20 | 19 | 18 | | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | | 9 | 8 | 7 | 6 | 5 | | 4 | 3 | 2 | 1 | 0 |
| bits | | name | | | | | | s/w | | | h/w | | | | default | | | | | description | | | | | | | | | | | | | | | | | | | | |
| 31:0 | | field | | | | | | rw | | | ro | | | | 0x0 | | | | | **{hdl\_path= upper\_limit\_field\_q}** | | | | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.4 lower\_limit | | | | | | | | | | | | | | | | lower\_limit | | | | | | | | | | | | | reg32 | | | | | | 0x000000c | | | | | |
| offset | | | | | | |  | | | | | | | | | external | | | | | | |  | | | | | | default | | | | | | 0x0 | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| block[@name='stopwatch']/reg[@name='lower\_limit'] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| The value written in this register is used to check the value in the "value" register which can't be lower than lower\_limit value. It has r/w permission under normal condition. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | | 29 | 28 | 27 | 26 | | | 25 | 24 | | 23 | 22 | 21 | | | 20 | 19 | 18 | | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | | 9 | 8 | 7 | 6 | 5 | | 4 | 3 | 2 | 1 | 0 |
| bits | | name | | | | | | s/w | | | h/w | | | | default | | | | | description | | | | | | | | | | | | | | | | | | | | |
| 31:0 | | field | | | | | | rw | | | ro | | | | 0x0 | | | | | **{hdl\_path= lower\_limit\_field\_q}** | | | | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.5 csr | | | | | | | | | | | | | | | | csr | | | | | | | | | | | | | reg32 | | | | | | 0x0000010 | | | | | |
| offset | | | | | | |  | | | | | | | | | external | | | | | | |  | | | | | | default | | | | | | 0x0 | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| block[@name='stopwatch']/reg[@name='csr'] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| This is control and status register. It consists of 4 fields with different widths and r/w permission. Stride field which is 4 bit wide is used to increment the value register when the value of updown field is 1 and decrement the value register when its value is 0. The value in the upper\_limit\_reached and lower\_limit\_reached is used to check the range limit of the value register. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | | 29 | 28 | 27 | 26 | | | 25 | 24 | | 23 | 22 | 21 | | | 20 | 19 | 18 | | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | | 9 | 8 | 7 | 6 | 5 | | 4 | 3 | 2 | 1 | 0 |
| bits | | name | | | | | | s/w | | | h/w | | | | default | | | | | description | | | | | | | | | | | | | | | | | | | | |
| 3:0 | | stride | | | | | | rw | | | ro | | | | 0x0 | | | | | Stride field**{hdl\_path= csr\_stride\_q}** | | | | | | | | | | | | | | | | | | | | |
| 4 | | updown | | | | | | rw | | | ro | | | | 0x0 | | | | | Up/Down field**{hdl\_path= csr\_updown\_q}** | | | | | | | | | | | | | | | | | | | | |
| 5 | | upper\_limit\_reached | | | | | | rw | | | ro | | | | 0x0 | | | | | Indicates that the upper limit has reached{**hdl\_path= csr\_upper\_limit\_reached\_q}** | | | | | | | | | | | | | | | | | | | | |
| 6 | | lower\_limit\_reached | | | | | | rw | | | ro | | | | 0x0 | | | | | Indicates that the lower limit has reached**{hdl\_path= csr\_lower\_limit\_reached\_q}** | | | | | | | | | | | | | | | | | | | | |
| 31:7 | | reserved\_field | | | | | | rw | | | ro | | | | 0x00 | | | | | reserved field**{hdl\_path= csr\_reserved\_field\_q}** | | | | | | | | | | | | | | | | | | | | |