AXIS2AXI

Блок, который принимает 5 stream-потоков и передает их по AXI.

Интерфейс блока:

input clk\_i ,

input aresetn\_i ,

//apb slave

input apb\_clk\_i ,

input apb\_paddr\_i ,

input apb\_psel\_i ,

input apb\_penable\_i ,

input apb\_pwrite\_i ,

input apb\_strb\_i ,

input apb\_pwdata\_i ,

output apb\_pready\_o ,

output apb\_prdata\_o ,

//first axis

input [31:0] axis\_one\_tdata\_i ,

input [3:0] axis\_one\_tkeep\_i ,

input axis\_one\_tlast\_i ,

input axis\_one\_tuser\_i ,

input axis\_one\_tvalid\_i ,

output axis\_one\_tready\_o ,

//second axis

input [31:0] axis\_two\_tdata\_i ,

input [3:0] axis\_two\_tkeep\_i ,

input axis\_two\_tlast\_i ,

input axis\_two\_tuser\_i ,

input axis\_two\_tvalid\_i ,

output axis\_two\_tready\_o ,

//third axis

input [31:0] axis\_three\_tdata\_i ,

input [3:0] axis\_three\_tkeep\_i ,

input axis\_three\_tlast\_i ,

input axis\_three\_tuser\_i ,

input axis\_three\_tvalid\_i ,

output axis\_three\_tready\_o ,

//fourd axis

input [63:0] axis\_four\_tdata\_i ,

input [7:0] axis\_four\_tkeep\_i ,

input axis\_four\_tlast\_i ,

input axis\_four\_tuser\_i ,

input axis\_four\_tvalid\_i ,

output axis\_four\_tready\_o ,

//fifth axis

input [255:0] axis\_five\_tdata\_i ,

input [31:0] axis\_five\_tkeep\_i ,

input axis\_five\_tlast\_i ,

input axis\_five\_tuser\_i ,

input axis\_five\_tvalid\_i ,

output axis\_five\_tready\_o ,

//axi master

output [3:0] m\_axi\_awid ,

output [31:0] m\_axi\_awaddr ,

output [7:0] m\_axi\_awlen ,

output [2:0] m\_axi\_awsize ,

output [1:0] m\_axi\_awburst ,

input m\_axi\_awready ,

output m\_axi\_awvalid ,

output [127:0] m\_axi\_wdata ,

output [15:0] m\_axi\_wstrb ,

output m\_axi\_wlast ,

output m\_axi\_wvalid ,

input m\_axi\_wready ,

input [1:0] m\_axi\_bresp ,

input m\_axi\_bvalid ,

output m\_axi\_bready

Где

apb\_\* apb интерфейс

aresetn\_i – синхронный сброс, активный уровень ноль для логики на частоте работы блока

clk\_i – 125 МГц частота работы блока

axis\_\* - входные stream потоки

m\_axi\_\* - AXI Master (write only)

Внутренние регистры

Регистры доступны по apb

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес | Description | Default value after reset | Access | Описание |
| 0x00 | ID\_DEVICE | 0xA314BD1A | RO | ID устройства |
| 0x04 | VERSION | 0x01 | RO | Версия модуля |
| 0x08 | NUMBER\_OF\_STREAMS | 0x5 | RO | Количество потоков |
| 0x0C | HEADER\_SIZE | 0x20 | RO | Размер header кадра в байтах |
| 0x10 | RESERVED | - | - | - |
| 0x14 | HSIZE\_VSIZE\_1 | 0x07B004CB | RO | hzise и vsize первого потока |
| 0x18 | BPP\_STRIDE\_1 | 0x00020800 | RO | bpp и stride первого потока |
| 0x1С | TYPE\_1 | 0x20363159 | RO | type первого потока |
| 0x20 | HSIZE\_VSIZE\_2 | 0x07B004CB | RO | hzise и vsize второго потока |
| 0x24 | BPP\_STRIDE\_2 | 0x00020800 | RO | bpp и stride второго потока |
| 0x28 | TYPE\_2 | 0x32525942 | RO | type второго потока |
| 0x2С | HSIZE\_VSIZE\_3 | 0x03D80265 | RO | hzise и vsize третьего потока |
| 0x30 | BPP\_STRIDE\_3 | 0x00040400 | RO | bpp и stride третьего потока |
| 0x34 | TYPE\_3 | 0x20424752 | RO | type третьего потока |
| 0x38 | HSIZE\_VSIZE\_4 | 0x02000080 | RO | hzise и vsize четвертого потока |
| 0x3C | BPP\_STRIDE\_4 | 0x00080200 | RO | bpp и stride четвертого потока |
| 0x40 | TYPE\_4 | 0x305A4741 | RO | type четвертого потока |
| 0x44 | HSIZE\_VSIZE\_5 | 0x00800200 | RO | hzise и vsize пятого потока |
| 0x48 | BPP\_STRIDE\_5 | 0x00200080 | RO | bpp и stride пятого потока |
| 0x4C | TYPE\_5 | 0x31444741 | RO | type пятого потока |
| 0x50  –  0x7C | RESERVED | - | - | - |
| 0x80 | FRAME\_PTR\_0 | 0 | RW | Адрес первого байта памяти для 0-го кадра |
| 0x84 | FRAME\_PTR\_1 | 0 | RW | Адрес первого байта памяти для 1-го кадра |
| 0x88 | FRAME\_PTR\_2 | 0 | RW | Адрес первого байта памяти для 2-го кадра |
| 0x8C | FRAME\_PTR\_3 | 0 | RW | Адрес первого байта памяти для 3-го кадра |
| 0x90 | FRAME\_PTR\_4 | 0 | RW | Адрес первого байта памяти для 4-го кадра |
| 0x94 | FRAME\_PTR\_5 | 0 | RW | Адрес первого байта памяти для 5-го кадра |
| 0x98 | FRAME\_PTR\_6 | 0 | RW | Адрес первого байта памяти для 6-го кадра |
| 0x9C | FRAME\_PTR\_7 | 0 | RW | Адрес первого байта памяти для 7-го кадра |
| 0xA0  –  0xBC | RESERVED | - | - | - |
| 0xC0 | FRAME\_STATUS\_PTR | 0 | RW | Адрес первого байта памяти для статуса готовности кадров |
| 0xC4 | MASK\_ENABLE | 0 | RW | Регистр, разрешающая работу по потокам |
| 0xC8 | START | 0 | RW | Регистр, который разрешает работу блока |
| 0xCC | FRAME\_STATUS | 0 | RO | Регистр статуса готовности кадров |

**ID\_DEVICE**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| ID\_REG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

ID\_REG, 32 бит. Идентификатор APB-устройства.

**VERSION**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| VERSION\_REG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

VERSION\_REG, 32 бит. Версия модуля.

**NUMBER\_OF\_STREAMS**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| NUMBER\_OF\_STREAMS\_REG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

NUMBER\_OF\_STREAMS\_REG, 32 бит. Количество stream-потоков, которые принимает модуль.

**HSIZE\_VSIZE\_N**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| HSIZE\_N\_REG | | | | | | | | | | | | | | | | VSIZE\_N\_REG | | | | | | | | | | | | | | | |

HSIZE\_N\_REG, 16 бит. hsize N-го потока.

VSIZE\_N\_REG, 16 бит. vsize N-го потока.

где N = 1…NUMBER\_OF\_STREAMS

**BPP\_STRIDE\_N**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| BPP\_N\_REG | | | | | | | | | | | | | | | | STRIDE\_N\_REG | | | | | | | | | | | | | | | |

BPP\_N\_REG, 16 бит. bpp N-го потока.

STRIDE\_N\_REG, 16 бит. stride N-го потока.

где N = 1…NUMBER\_OF\_STREAMS

**TYPE\_N**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| TYPE\_N\_REG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

TYPE\_N\_REG, 32 бит. type N-го потока.

где N = 1…NUMBER\_OF\_STREAMS

**FRAME\_PTR\_N**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| FRAME\_PTR\_REG\_N | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

FRAME\_PTR\_REG\_N, 32 бита. Значение регистра определяет адрес первого байта области памяти, куда блок будет писать N-й кадр.

где N = 0…7

**FRAME\_STATUS\_PTR**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| FRAME\_STATUS\_PTR\_REG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

FRAME\_STATUS\_PTR\_REG, 32 бит. Значение регистра определяет адрес первого байта области памяти, куда блок будет писать информацию о готовности кадров. Пишет в память значение регистра FRAME\_STATUS, чтобы исключить чтение.

**MASK\_ENABLE**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | MASK\_REG | | | | |

MASK\_REG, 5 бит. Значение регистра разрешает работу блока по потоку. Например, 0 бит регистра разрешает принимать данные по 1-му stream потоку и передавать их в AXI-master. 1-й бит разрешает принимать данные по 2-му потоку и так далее. ”1” – разрешает работу, “0” – запрещает.

**START**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | START\_REG |

START\_REG, 1 бит. Разрешает работу блока. Обновление всех рабочих внутренних регистров блока происходит по положительному фронту это регистра.

**FRAME\_STATUS**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | FRAME\_STATUS\_REG | | | | | | | |

FRAME\_STATUS\_REG, 8 бит. Значение регистра говорит о готовности кадров. Если единица, то кадр полностью лежит в памяти. В дальнейшем данный регистр может быть скрыт от пользователя.

Настройки блока

Данные настройки доступны в файле axis\_define.v

IF\_VALUE\_TRIG\_ONE – при записи во входной буфер 1-го потока кол-во слов больше чем данное значение, то создается AXI-транзакция

IF\_VALUE\_TRIG\_TWO – при записи во входной буфер 2-го потока кол-во слов больше чем данное значение, то создается AXI-транзакция

IF\_VALUE\_TRIG\_THREE – при записи во входной буфер 3-го потока кол-во слов больше чем данное значение, то создается AXI-транзакция

IF\_VALUE\_TRIG\_FOUR – при записи во входной буфер 4-го потока кол-во слов больше чем данное значение, то создается AXI-транзакция

IF\_VALUE\_TRIG\_FIVE – при записи во входной буфер 5-го потока кол-во слов больше чем данное значение, то создается AXI-транзакция

OFFSET\_ONE – оффсет в кортеже для 2-го потока

OFFSET\_TWO – оффсет в кортеже для 3-го потока

OFFSET\_THREE – оффсет в кортеже для 4-го потока

OFFSET\_FOUR – оффсет в кортеже для 5-го потока

OFFSET\_HEADER – оффсет в кортеже для header