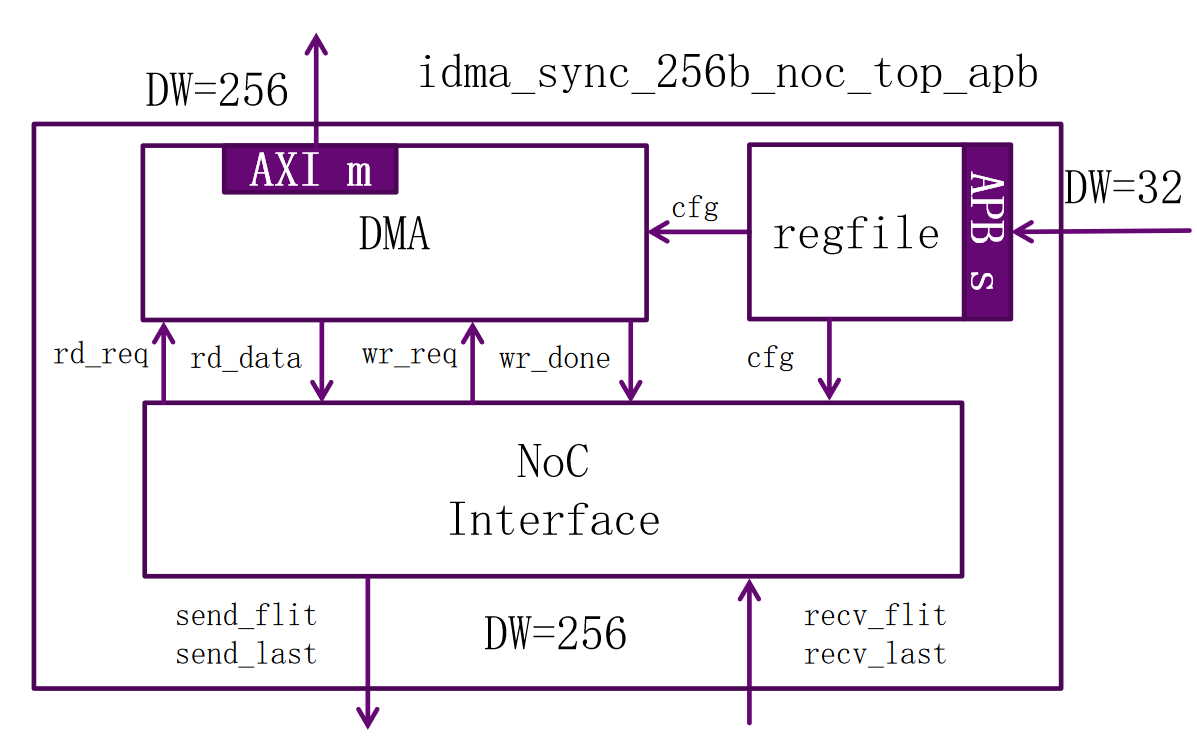
dma设计文档

|  |  |
| --- | --- |
| 设计人员 | 林宥旭 |
| 修改日期 | 2024.5.7 |

# 功能

|  |  |
| --- | --- |
| DMA功能 | 只支持读通道 |
| 位宽256bit |
| 同步设计，无跨时钟设计 |
| 最大burst length为16 |
| 配置接口 | 支持APB4协议 |
| 位宽32bit |
| 与NoC的接口 | 与一个NoC节点连接 |
| 位宽256bit |
| 不支持outstanding |
| 带有last信号 |
| 中断 | 支持中断状态的开启/读取/清除 |

# 架构



# 接口和寄存器

## 接口

|  |  |  |  |
| --- | --- | --- | --- |
| 信号名 | 位宽 | 方向 | 含义 |
| **clock/reset** | | | |
| aclk | 1 | input | 时钟 |
| aresetn | 1 | input | 复位，低电平有效 |
| **APB config slave 32** | | | |
| apb\_PADDR | 12 | input | 参考APB4协议 |
| apb\_PSEL | 1 | input | 参考APB4协议 |
| apb\_PENABLE | 1 | input | 参考APB4协议 |
| apb\_PREADY | 1 | output | 参考APB4协议 |
| apb\_PWRITE | 1 | input | 参考APB4协议 |
| apb\_PSTRB | 4 | input | 参考APB4协议 |
| apb\_PPROT | 3 | input | 参考APB4协议 |
| apb\_PWDATA | 32 | input | 参考APB4协议 |
| apb\_PRDATA | 32 | output | 参考APB4协议 |
| apb\_PSLVERR | 1 | output | 参考APB4协议 |
| **AIX master of DMA** | | | |
| arvalid | 1 | output | 参考AXI4协议 |
| arid | 4 | output | 参考AXI4协议 |
| araddr | 32 | output | 参考AXI4协议 |
| arlen | 8 | output | 参考AXI4协议 |
| arsize | 3 | output | 参考AXI4协议 |
| arburst | 2 | output | 参考AXI4协议 |
| arlock | 1 | output | 参考AXI4协议 |
| arcache | 4 | output | 参考AXI4协议 |
| arprot | 3 | output | 参考AXI4协议 |
| arready | 1 | input | 参考AXI4协议 |
| rvalid | 1 | input | 参考AXI4协议 |
| rid | 4 | input | 参考AXI4协议 |
| rlast | 1 | input | 参考AXI4协议 |
| rdata | 128 | input | 参考AXI4协议 |
| rresp | 2 | input | 参考AXI4协议 |
| rready | 1 | output | 参考AXI4协议 |
| awvalid | 1 | output | 参考AXI4协议 |
| awid | 4 | output | 参考AXI4协议 |
| awaddr | 32 | output | 参考AXI4协议 |
| awlen | 4 | output | 参考AXI4协议 |
| awsize | 3 | output | 参考AXI4协议 |
| awburst | 2 | output | 参考AXI4协议 |
| awlock | 2 | output | 参考AXI4协议 |
| awcache | 4 | output | 参考AXI4协议 |
| awprot | 3 | output | 参考AXI4协议 |
| awready | 1 | input | 参考AXI4协议 |
| wvalid | 1 | output | 参考AXI4协议 |
| wid | 4 | output | 参考AXI4协议 |
| wlast | 1 | output | 参考AXI4协议 |
| wdata | 256 | output | 参考AXI4协议 |
| wstrb | 32 | output | 参考AXI4协议 |
| wready | 1 | input | 参考AXI4协议 |
| bvalid | 1 | input | 参考AXI4协议 |
| bid | 4 | input | 参考AXI4协议 |
| bresp | 2 | input | 参考AXI4协议 |
| bready | 1 | output | 参考AXI4协议 |
| **send to NoC** | | | |
| send\_valid | 1 | output | 发送有效信号 |
| send\_last | 1 | output | 发送last信号 |
| send\_flit | 32 | output | 发送数据 |
| send\_ready | 1 | input | 发送端ready信号 |
| **receive from NoC** | | | |
| recv\_valid | 1 | input | 接收端有效信号 |
| recv\_last | 1 | input | 接收last信号 |
| recv\_flit | 32 | input | 接收数据 |
| recv\_ready | 1 | output | 接收端ready信号 |
| **interrupt** | | | |
| interrupt | 1 | output | 中断信号 |

## 寄存器

| **AddressOffset** | **RegName** | **Description** | **Width** | **Section** | **FieldName** | **R/W** | **Reset value** | **Field-Description** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0x0000 | IDMA\_RD\_CFG\_EN | Enable DMA Read | 32 | [31:1] | -- | NA | 31'b0 | Reserved |
| [0] | rd\_cfg\_en | RW | 1'b0 | read config enable |
| 0x0004 | IDMA\_RD\_AFIFO\_CONTROL | Init DMA Read Address FIFO | 32 | [31:1] | -- | NA | 31'b0 | Reserved |
| [0] | rd\_afifo\_init | W1P | 1'b0 | read addr fifo init |
| 0x0008 | IDMA\_RD\_DFIFO\_CONTROL | Control DMA Read Data FIFO | 32 | [31:24] | -- | NA | 8'b0 | Reserved |
| [23:16] | rd\_cfg\_dfifo\_thd | RW | 8'b0 | read data fifo threshold |
| [15:1] | -- | NA | 15'b0 | Reserved |
| [0] | rd\_dfifo\_init | W1P | 1'b0 | read data fifo init |
| 0x000C | IDMA\_RD\_CTRL0 | Control DMA Read No.0 | 32 | [31:8] | -- | NA | 24'b0 | Reserved |
| [7] | rd\_cfg\_resi\_mode | RW | 1'b0 | resi mode |
| [6] | rd\_cfg\_arvld\_hold\_en | RW | 1'b0 | enable hold ar valid |
| [5] | rd\_cfg\_cross4k\_en | RW | 1'b0 | enable cross 4k |
| [4] | rd\_cfg\_outstd\_en | RW | 1'b0 | enable outstanding |
| [3:0] | rd\_cfg\_outstd | RW | 4'b0 | max outstanding num |
| 0x0010 | IDMA\_RD\_RESI\_FMAP\_A | Fmap A Address of DMA Read at Resi Mode | 32 | [31:0] | rd\_cfg\_resi\_fmap\_a\_addr | RW | 32'b0 | resi mode A address |
| 0x0014 | IDMA\_RD\_RESI\_FMAP\_B | Fmap B Address of DMA Read at Resi Mode | 32 | [31:0] | rd\_cfg\_resi\_fmap\_b\_addr | RW | 32'b0 | resi mode B address |
| 0x0018 | IDMA\_RD\_RESI\_ADDR\_GAP | Address Gap of DMA Read at Resi Mode | 32 | [31:0] | rd\_cfg\_resi\_addr\_gap | RW | 32'b0 | resi mode address gap |
| 0x001C | IDMA\_RD\_RESI\_LOOP\_NUM | Loop Number of DMA Read at Resi Mode | 32 | [31:0] | rd\_cfg\_resi\_loop\_num | RW | 32'b0 | resi mode loop number |
| 0x0020 | IDMA\_WR\_CFG\_EN | Enable DMA Write | 32 | [31:1] | -- | NA | 31'b0 | Reserved |
| [0] | wr\_cfg\_en | RW | 1'b0 | write config enable |
| 0x0024 | IDMA\_WR\_AFIFO\_CONTROL | Init DMA Write Address FIFO | 32 | [31:1] | -- | NA | 31'b0 | Reserved |
| [0] | wr\_afifo\_init | W1P | 1'b0 | write addr fifo init |
| 0x0028 | IDMA\_WR\_DFIFO\_CONTROL | Control DMA Write Data FIFO | 32 | [31:24] | -- | NA | 8'b0 | Reserved |
| [23:16] | wr\_cfg\_dfifo\_thd | RW | 8'b0 | write data fifo threshold |
| [15:1] | -- | NA | 15'b0 | Reserved |
| [0] | wr\_dfifo\_init | W1P | 1'b0 | write data fifo init |
| 0x002C | IDMA\_WR\_CTRL0 | Control DMA Write No.0 | 32 | [31:9] | -- | NA | 23'b0 | Reserved |
| [8] | wr\_cfg\_strb\_force | RW | 1'b0 | force strb to ffffffff |
| [7] | wr\_cfg\_arvld\_hold\_olen\_en | RW | 1'b0 | wr enable hold aw len |
| [6] | wr\_cfg\_arvld\_hold\_en | RW | 1'b0 | wr enable hold aw valid |
| [5] | wr\_cfg\_cross4k\_en | RW | 1'b0 | wr enable cross 4k |
| [4] | wr\_cfg\_outstd\_en | RW | 1'b0 | wr enable outstanding |
| [3:0] | wr\_cfg\_outstd | RW | 4'b0 | wr max outstanding num |
| 0x0030 | IDMA\_BASE\_ADDR0 | Base Addr 0 | 32 | [31:0] | base\_addr\_0 | RW | 32'b0 | Base Addr 0 |
| 0x0034 | IDMA\_BASE\_ADDR1 | Base Addr 1 | 32 | [31:0] | base\_addr\_1 | RW | 32'b0 | Base Addr 1 |
| 0x0038 | IDMA\_BASE\_ADDR2 | Base Addr 2 | 32 | [31:0] | base\_addr\_2 | RW | 32'b0 | Base Addr 2 |
| 0x003C | IDMA\_BASE\_ADDR3 | Base Addr 3 | 32 | [31:0] | base\_addr\_3 | RW | 32'b0 | Base Addr 3 |
| 0x0040 | IDMA\_BASE\_ADDR4 | Base Addr 4 | 32 | [31:0] | base\_addr\_4 | RW | 32'b0 | Base Addr 4 |
| 0x0044 | IDMA\_BASE\_ADDR5 | Base Addr 5 | 32 | [31:0] | base\_addr\_5 | RW | 32'b0 | Base Addr 5 |
| 0x0048 | IDMA\_BASE\_ADDR6 | Base Addr 6 | 32 | [31:0] | base\_addr\_6 | RW | 32'b0 | Base Addr 6 |
| 0x004C | IDMA\_BASE\_ADDR7 | Base Addr 7 | 32 | [31:0] | base\_addr\_7 | RW | 32'b0 | Base Addr 7 |
| 0x0050 | IDMA\_BASE\_ADDR8 | Base Addr 8 | 32 | [31:0] | base\_addr\_8 | RW | 32'b0 | Base Addr 8 |
| 0x0054 | IDMA\_BASE\_ADDR9 | Base Addr 9 | 32 | [31:0] | base\_addr\_9 | RW | 32'b0 | Base Addr 9 |
| 0x0058 | IDMA\_BASE\_ADDR10 | Base Addr 10 | 32 | [31:0] | base\_addr\_10 | RW | 32'b0 | Base Addr 10 |
| 0x005C | IDMA\_BASE\_ADDR11 | Base Addr 11 | 32 | [31:0] | base\_addr\_11 | RW | 32'b0 | Base Addr 11 |
| 0x0060 | IDMA\_RD\_CNT | DMA read in Count | 32 | [31:16] | -- | NA | 16'b0 | Reserved |
| [15:0] | debug\_dma\_rd\_in\_cnt | RO | 16'b0 | DMA read in Count |
| 0x0064 | IDMA\_WR\_CNT | DMA write out Count | 32 | [31:16] | -- | NA | 16'b0 | Reserved |
| [15:0] | debug\_dma\_wr\_out\_cnt | RO | 16'b0 | DMA write out Count |
| 0x0068 | IDMA\_INTR\_INT\_RAW | Interrupt Raw status Register set when event clear when write 1 | 32 | [31:2] | -- | NA | 30'b0 | Reserved |
| [1] | rd\_done\_intr\_raw | W1C | 1'b0 | raw, default 0 |
| [0] | wr\_done\_intr\_raw | W1C | 1'b0 | raw, default 0 |
| 0x006C | IDMA\_INTR\_INT\_MASK | Interrupt Mask Register 1: int off 0: int open default 1, int off | 32 | [31:2] | -- | NA | 30'b0 | Reserved |
| [1] | rd\_done\_intr\_mask | RW | 1'h1 | mask, default 1, int off |
| [0] | wr\_done\_intr\_mask | RW | 1'h1 | mask, default 1, int off |
| 0x0070 | IDMA\_INTR\_INT\_STATUS | Interrupt status Register status = raw && (!mask) | 32 | [31:2] | -- | NA | 30'b0 | Reserved |
| [1] | rd\_done\_intr\_status | RO | 1'b0 | stauts default 0 |
| [0] | wr\_done\_intr\_status | RO | 1'b0 | stauts default 0 |

# 工作流程

## DMA

读功能：根据需求配置IDMA\_RD\_CTRL0寄存器，然后配置IDMA\_RD\_CFG\_EN寄存器让读请求使能。当NoC接口接收到读请求时，DMA开始对外发送读请求。

写功能：根据需求配置IDMA\_WR\_CTRL0寄存器，然后配置IDMA\_WR\_CFG\_EN寄存器让写请求使能。当NoC接口接收到写请求时，DMA开始对外发送写请求。

## 与NoC交互

配置寄存器IDMA\_BASE\_ADDR0~IDMA\_BASE\_ADDR11，表示DMA根据不同节点请求加上的基地址。

读的过程为：接收第一条flit（低128bit有效），其中第0位表示读写（0为读，1为写），[5:2]为源节点坐标，[71:56]为地址，[91:72]为长度。接收第二条flit。DMA根据第一条的信息对外发送读请求，然后接收返回的读数据。发送端第一条为接收第一条的改写第0位为1（表示写），其余位不变；发送的第二条为接收的第二条；后续发送的即为读数据，在最后一个读数据把last拉高。

写的过程为：接收第一条flit（低128bit有效），其中第0位表示读写（0为读，1为写），[5:2]为源节点坐标，[71:56]为地址，[91:72]为长度。接收第二条flit。后续接收的即为要写的数据。DMA根据第一条的信息对外发送写请求，然后接收返回的写响应。发送端第一条为接收第一条的改写第0位为1（表示写），第1位为1（表示写结束），同时把last拉高，因为写请求结束后只会返回写结束的信号。