

**Техническое описание**

**IP-блока контроллера  
Ethernet Gigabit MAC (GMAC),**

**ver.1.2**

Основные характеристики:

- поддержка режимов работы 10/100/1000 Мбит/с;
- соответствие 802.3 спецификации;- интерфейс связи с микросхемой Phy — MII/GMII;
- поддержка полудуплексного обмена в режимах 10, 100 Мбит/с;
- AXI4 32p DMA интерфейс по каждому из каналов передачи данных Rx, Tx;
- поддержка single и burst режимов AXI4;
- отдельное настраиваемое формирование прерываний по каждому из каналов Rx, Tx;- AXI4-lite 32p slave интерфейс для настройки режимов работы и управления;
- поддержка генерации и приема пакетов pause;
- поддержка автоподстановки source MAC address в формируемые пакеты;
- возможность отключения проверок CRC, destination MAC address;
- наличие внутренней тестовой петли;
- накопление статистической информации.

1. Условное графическое обозначение.

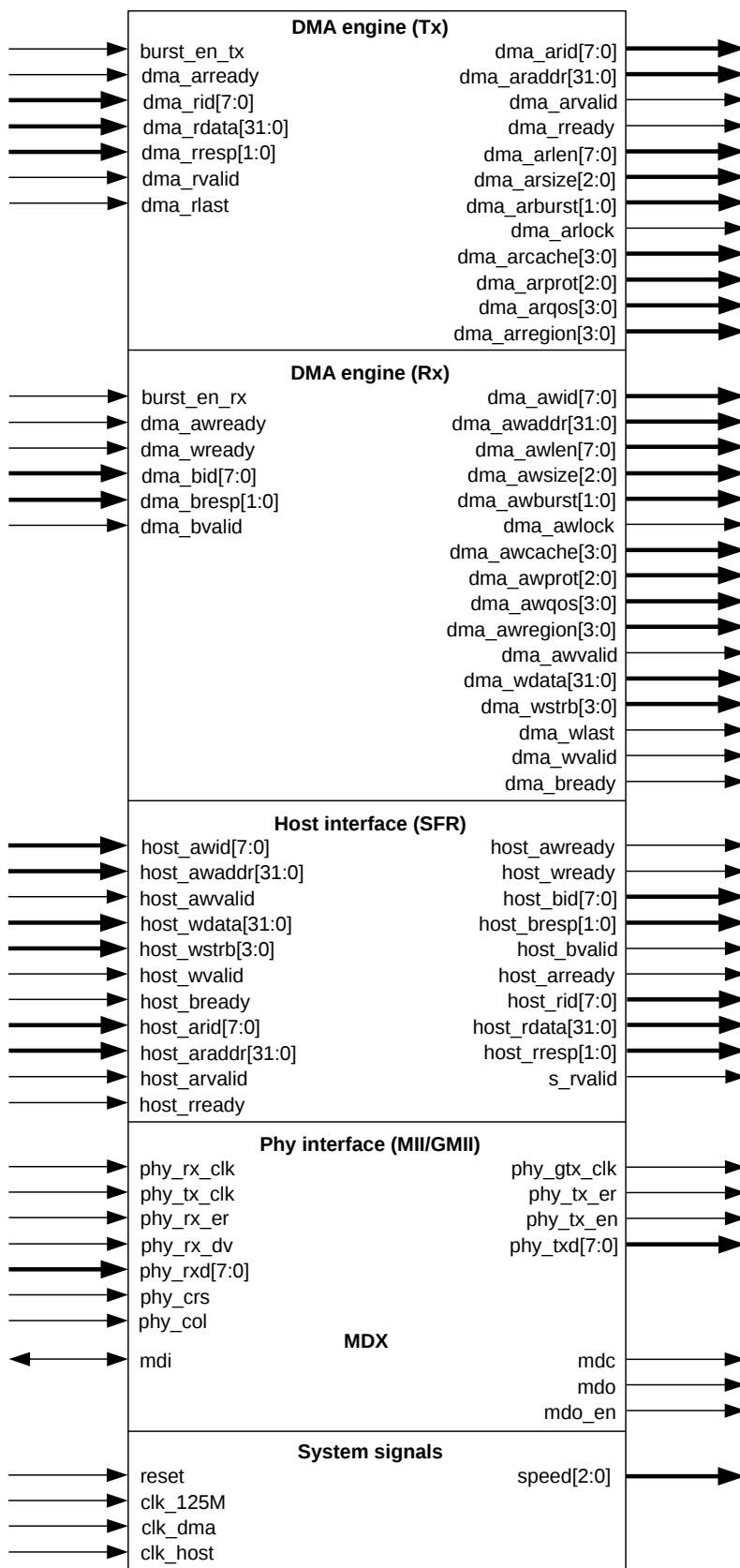


Рисунок 1 — УГО блока GMAC

## 2. Описание портов

Таблица 1 — Описание портов блока GMAC

Имя порта	Направление	Примечания
<b>Системные сигналы</b>		
reset_sys	ВХ	
clk_dma	ВХ	
clk_host	ВХ	
clk_250	ВХ	
speed[2:0]	ВЫХ	
irq_tx	ВЫХ	
irq_rx	ВЫХ	
<b>DMA канал записи (AXI4)</b>		
dma_awid[7:0]	ВЫХ	
dma_awaddr[31:0]	ВЫХ	
dma_awvalid	ВЫХ	
dma_awready	ВХ	
dma_wdata[31:0]	ВЫХ	
dma_wstrb[3:0]	ВЫХ	
dma_wvalid	ВЫХ	
dma_wready	ВХ	
dma_bid[7:0]	ВХ	
dma_bresp[1:0]	ВХ	
dma_bvalid	ВХ	
dma_bready	ВЫХ	
<b>Сигналы пакетного режима DMA канала записи</b>		
dma_wlast	ВЫХ	
dma_awlen[7:0]	ВЫХ	
dma_awsz[2:0]	ВЫХ	
dma_awburst[1:0]	ВЫХ	
dma_awlock	ВЫХ	
dma_awcache[3:0]	ВЫХ	
dma_awprot[2:0]	ВЫХ	
dma_awqos[3:0]	ВЫХ	
dma_awregion[3:0]	ВЫХ	
<b>DMA канал чтения (AXI4)</b>		
dma_arid[7:0]	ВЫХ	
dma_araddr[31:0]	ВЫХ	
dma_arvalid	ВЫХ	
dma_arready	ВХ	

dma_rid[7:0]	ВХ	
dma_rdata[31:0]	ВХ	
dma_rresp	ВХ	
dma_rvalid	ВХ	
dma_rready	ВЫХ	
<b>Сигналы пакетного режима DMA канала чтения</b>		
dma_rlast	ВХ	
dma_arlen[7:0]	ВЫХ	
dma_arsize[2:0]	ВЫХ	
dma_arburst[1:0]	ВЫХ	
dma_arlock	ВЫХ	
dma_arcache[3:0]	ВЫХ	
dma_arprot[2:0]	ВЫХ	
dma_arqos[3:0]	ВЫХ	
dma_arregion[3:0]	ВЫХ	
<b>Хост интерфейс (AXI4-lite slave)</b>		
host_awid[7:0]	ВХ	
host_awaddr[31:0]	ВХ	
host_awvalid	ВХ	
host_awready	ВЫХ	
host_wdata[31:0]	ВХ	
host_wstrb[3:0]	ВХ	
host_wvalid	ВХ	
host_wready	ВЫХ	
host_bid[7:0]	ВЫХ	
host_bresp[1:0]	ВЫХ	
host_bvalid	ВЫХ	
host_bready	ВХ	
host_arid[7:0]	ВХ	
host_araddr[31:0]	ВХ	
host_arvalid	ВХ	
host_arready	ВЫХ	
host_rid[7:0]	ВЫХ	
host_rdata[31:0]	ВЫХ	
host_rresp[1:0]	ВЫХ	
host_rvalid	ВЫХ	
host_rready	ВХ	
<b>МП/ГМП интерфейс</b>		
Gtx_clk	ВЫХ	

Rx_clk	ВХ	
Tx_clk	ВХ	
Tx_er	ВЫХ	
Tx_en	ВЫХ	
Txd[7:0]	ВЫХ	
Rx_er	ВХ	
Rx_dv	ВХ	
Rxd[7:0]	ВХ	
Crs	ВХ	
Col	ВХ	
<b>MDIO интерфейс</b>		
Mdo	ВЫХ	
MdoEn	ВЫХ	
Mdi	ВХ	
Mdc	ВЫХ	

### 3. Блок-схема

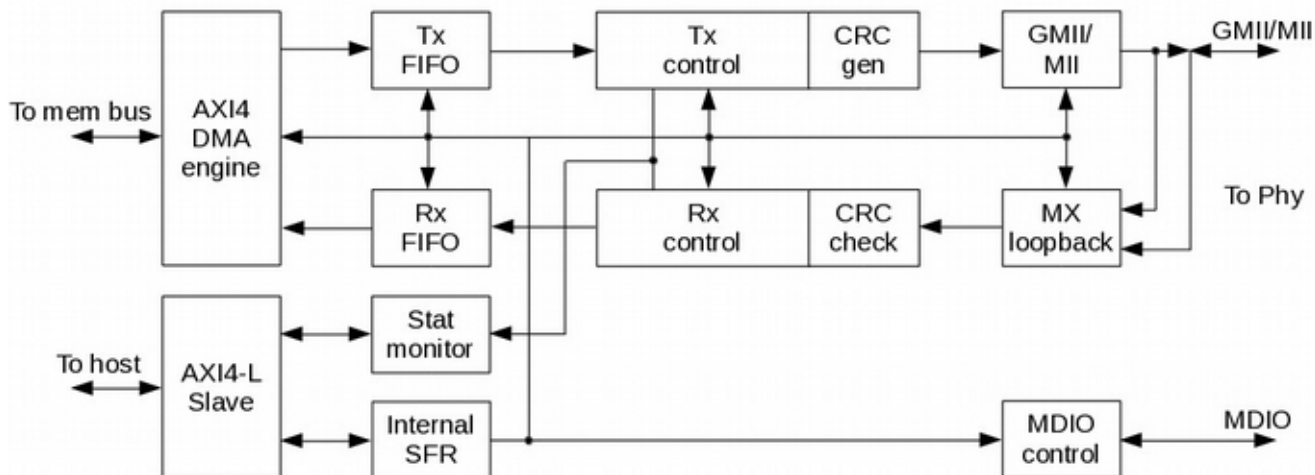


Рисунок 2 — Блок-схема контроллера Gigabit Ethernet MAC

#### 3.1. Описание составных блоков:

**AXI4 DMA Engine** — мастер интерфейс шины AXI4 для подключения к памяти, напрямую либо через AXI4-коммутатор. Адресация 32р, шина данных 32р. Поддерживается блочная (burst) передача. Важно отметить, что подключаемое slave устройство поддерживало одновременный обмен по каналам чтения и записи, в противном случае при подключении к AXI4 DMA engine необходимо подключать 2 независимых slave устройства по одному на каждый канал записи и чтения.

**AXI4-L Slave** — AXI4-lite slave контроллер для подключения host устройства. Адресация 32р, шина данных 32р. Используется для настройки внутренних регистров GMAC, Internal SFR, для обмена данными с микросхемой Phy по MDIO интерфейсу, а также чтения статистической информации из блока Stat monitor.

**Tx FIFO** — FIFO Tx канала GMAC, размер 2048 байт.

**Rx FIFO** — FIFO Rx канала GMAC, размер 2048 байт, с функцией быстрой очистки

**Stat monitor** — счетчики статистических данных. См. Таблицу 3

**Internal SFR** — внутренние регистры настройки GMAC. См. Таблицу 2

**Tx control** — логика управления Tx каналом GMAC

**Rx control** — логика управления Rx каналом GMAC

**GMII/MII** — блок реализации интерфейсов GMII/MII для обмена данными с микросхемой Phy

**MX loopback** — мультиплексор включения тестовой петли обратной связи

**MDIO control** — блок реализации интерфейса MDIO для обмена данными с микросхемой Phy для настройки режимов ее работы, чтения статистической информации.

#### 4. Адресное пространство

Таблица 2 — Регистры:

Адрес	Регистр	Напр.	Описание	По-умолчанию
0x000	Tx_Hwmark	R/w	Значения, определяющие верхнюю (Tx_Hwmark) и нижнюю (Tx_Lwmark) границы уровня заполнения Fifo Tx. Если объем данных в Fifo достигает нижней границы, логика управления выдачей начинает вычитывать данные из Fifo и выдавать их в Phy. В зависимости от достижения верхней или нижней границы происходит формирование сигнала Tx_mac_wt. Когда достигается верхняя граница, сигнал Tx_mac_wt устанавливается в 0, сообщая блоку DMA Tx о необходимости приостановить загрузку в Fifo. Когда при выводе достигается нижняя граница, Tx_mac_wt устанавливается в 1 и блок DMA Tx может приступить к продолжению загрузки в Fifo. Шаг изменения значения = 64 байта.	32'h0000001e
0x004	Tx_Lwmark	R/w		32'h00000019
0x008	pause_frame_send_en	R/w	Включение логики передачи для отправки кадра PAUSE	32'h00000000
0x00c	pause_quanta_set	R/w	Определяет значение квантов задержки в кадре PAUSE. Один квант равен задержке при передаче 512 битного пакета	32'h00000000
0x010	IFGset	R/w	Inter Frame Gap — задержка между кадрами в тактах clk125. По IEEE802.3, минимальное значение IFG составляет 96 бит. Если системная тактовая частота 125 МГц, а ширина данных 8 бит, то интервал между пакетами составляет не менее 18 тактов	32'h00000012
0x014	FullDuplex	R/w	Переключатель режимов FullDuplex/HalfDuplex. Работает только в режимах 100Мбит/с и 10Мбит/с. Если равен 1, установлен режим FullDuplex. В противном случае будут выполняться обнаружение коллизий, откат случайного интервала времени, повторно передаваться коллизийный пакет и некоторые другие HalfDuplex операции	32'h00000001
0x018	MaxRetry	R/w	Количество повторов передачи пакета при обнаружении коллизии в режиме HalfDuplex. Если после MaxRetry попыток передачи все же возникает коллизия, пакет будет отброшен	32'h00000002
0x01c	MAC_tx_addr_en	R/w	Установка в 1 этого регистра разрешает изменение MAC-адреса в Tx пакете значением из регистров MAC_tx_addr_xx	32'h00000000
0x020	MAC_tx_addr_10	R/w	0-й и 1-й байты MAC адреса: 32'h{16'hxxxx,byte1,byte0}	32'h00000001
0x024	MAC_tx_addr_32	R/w	2-й и 3-й байты MAC адреса 32'h{16'hxxxx,byte3,byte2}	32'h000006100
0x028	MAC_tx_addr_54	R/w	4-й и 5-й байты MAC адреса 32'h{16'hxxxx,byte5,byte4}	32'h000000d8



0x02c	tx_pause_en pause_apply_flag	R/w	tx_pause_en - этот бит разрешает реакцию на принятые пакеты PAUSE, расположен в [0] разряде. При приеме пакета PAUSE контроллер перейдет в режим паузы в на время в соответствии со значением квантов в принятом пакете. Один квант равен задержке при передаче 512 битного пакета  pause_apply_flag — флаг активности режима pause после приема пакета pause, расположен в [31] разряде, read only	32'h00000000
0x030	xon_cpu	R/w	Запуск передачи одного кадра PAUSE со значением pause_quanta_set	32'h00000000
0x034	xoff_cpu	R/w	Запуск передачи одного кадра PAUSE с нулевым значением кванта паузы, сообщая удаленному контроллеру Ethernet о необходимости выхода из режима паузы	32'h00000000
0x038	MAC_rx_addr_chk_en	R/w	Установка в 1 этого регистра разрешает проверку MAC-адреса в Rx пакете	32'h00000000
0x03c	MAC_rx_addr_10 <sup>1</sup>	R/w	0-й и 1-й байты MAC адреса: 32'h{16'hxxxx,byte1,byte0}	32'h00000001
0x040	MAC_rx_addr_32 <sup>1</sup>	R/w	2-й и 3-й байты MAC адреса 32'h{16'hxxxx,byte3,byte2}	32'h00006100
0x044	MAC_rx_addr_54 <sup>1</sup>	R/w	4-й и 5-й байты MAC адреса 32'h{16'hxxxx,byte5,byte4}	32'h000000d8
0x048	broadcast_filter_en	R/w	Установка в 1 включает фильтр ширококестельных пакетов	32'h00000000
0x04c	broadcast_bucket_depth	R/w	<i>Количество данных пакетов ширококестельной рассылки. С каждым принятым байтом ширококестельного пакета происходит уменьшение broadcast_bucket_depth на 1. Если вся глубина выбрана, все принятые ширококестельные пакеты будут игнорированы до следующей записи глубины или до окончания интервала приема ширококестельных пакетов broadcast_bucket_interval. Параметр задается на 1 меньше</i>	32'h00000000
0x050	broadcast_bucket_interval	R/w	Определяет интервал приема ширококестельных пакетов. На этом интервале будет принято не более broadcast_bucket_depth байт ширококестельных пакетов	32'h00000000
0x054	RX_APPEND_CRC	R/w	Установленный в 1 разрешает передачу значения FCS (Frame check sequence) или CRC пользовательскому приложению	32'h00000000
0x058	Rx_Hwmark <sup>2</sup>	R/w	Значения, определяющие верхнюю (Rx_Hwmark) и нижнюю (Rx_Lwmark) границы уровня заполнения Fifo Rx. Достижение той или иной границы влияет на формирование сигнала Rx_mac_га. Если получей один полный пакет либо достигнут верхний предел, сигнал Rx_mac_га будет установлен в 1. Когда будет	32'h0000001a
0x05c	Rx_Lwmark <sup>2</sup>	R/w		32'h00000010

			достигнута нижняя граница, сигнал Rx_max_га будет установлен в 0. Если в Fifo записан один полный пакет, сигнал Rx_max_га установится в 0 только после вычитывания блоком DMA Rx всего пакета. Шаг изменения значения = 64 байта.	
d0x060	CRC_chk_en	R/w	Разрешает проверку CRC принятого пакета. Пакеты с ошибочной CRC откидываются. Значение 0 отключает проверку CRC принятых пакетов	32'h00000001
0x064	RX_IFG_SET	R/w	Определяет интервал IFG между принимаемыми кадрами. Если разрыв между двумя полученными пакетами меньше RX_IFG_SET, второй пакет будет отброшен как недопустимый кадр	32'h00000012
0x068	RX_MAX_LENGTH	R/w	Задают ограничения максимального и минимального размера принятого пакета. Если размер пакета выходит за рамки RX_MAX_LENGTH и RX_MIN_LENGTH, он будет отброшен	32'h00002710
0x06c	RX_MIN_LENGTH	R/w		32'h00000040
0x070	<i>reserved</i>			32'h00000000
0x074	Rx_FIFO_clear	R/w	Очистка буфера FIFO приемника Rx	32'h00000000
0x078	xMII_COL	R	COL line MII interface status	32'h00000000
-	<i>reserved</i>			
0x084	Line_loop_en	R/w	Разрешение тестовой петли Tx->Rx. Выдаваемый в Phy пакет будет транслироваться на приемную часть	32'h00000000
0x088	Speed	R	Скорость: 3'b100 — 1000Мбит/с 3'b010 — 100Мбит/с 3'b001 — 10Мбит/с	32'h00000004
0x08c	MDIO_DividerHostClk	R/w	Коэффициент деления для формирования сигнала MDC из Host_Clk, [7:0]	32'h00000001
0x090	MDIO_WrCtrlData	R/w	Запись данных во внутренние регистры Phy: MDIO_WrCtrlData[15:0] — данные; MDIO_WrCtrlData[20:16] — адрес регистра Phy; MDIO_WrCtrlData[21] — запись в Phy команды MDIO_WrData[22] — запись в Phy/чтение из Phy	32'h00000000
0x094	MDIO_PhyAddress	R/w	MDIO_PhyAddr[4:0] — адрес Phy	32'h00000000
0x098	MDIO_NoPreamble	R/w	Обмен по MDIO с Phy без 32б преамбулы	32'h00000001
0x09c	MDIO_ReadStatusOp	R/w	Запуск операции чтения регистра статуса Phy	32'h00000000
0x0a0	MDIO_ScanStatusOp	R/w	Запуск операции циклического чтения регистра статуса Phy	32'h00000000
0x0a4	MDIO_StatusFlags	R	Состояние флагов: MDIO_StatusFlags[0] — «Busy», «1» если выполняется обмен с Phy; MDIO_StatusFlags[1] — «LinkFail»; MDIO_StatusFlags[2] — «PhyDataReady»	32'h00000000
0x0a8	MDIO_PhyData	R	MDIO_PhyData[15:0] — регистр полученных из Phy данных	32'h00000000

0x100	DMA_Addr_Start_Tx	R/w	Определяют границы кольцевого буфера Tx пакетов в системном ОЗУ. Указывается адрес в формате системной шины	32'h00000000
0x104	DMA_Addr_Finish_Tx	R/w		32'h00000000
0x108	DMA_Set_Tx	R/w	Параметры передачи: DMA_Set_Tx[31] — одиночный/множественный режим («1» - множественный); DMA_Set_Tx[30] — формирование прерывания после 1 пакета; DMA_Set_Tx[29] — формирование прерывания после всех пакетов; DMA_Set_Tx[28:12] - кол-во пакетов (до 128K); DMA_Set_Tx[11:0] - размер пакета (до 1500байт);	32'h00000000
0x10c	DMA_Addr_Start_Rx	R/w	Определяют границы кольцевого буфера Rx пакетов в системном ОЗУ. Указывается адрес в формате системной шины	32'h00000000
0x110	DMA_Addr_Finish_Rx	R/w		32'h00000000
0x114	DMA_Set_Rx	R/w	Параметры передачи: DMA_Set_Rx[31] — одиночный/множественный режим («1» - множественный); DMA_Set_Rx[30] — формирование прерывания после 1 пакета; DMA_Set_Rx[29] — формирование прерывания после всех пакетов; DMA_Set_Rx[28:12] - кол-во пакетов (до 128K); DMA_Set_Rx[11:0] - резерв	32'h00000000
0x118	DMA_Cmd	R/w	Регистр команд DMA engine: DMA_Cmd[3] — burst mode on Tx DMA, DMA_Cmd[2] — burst mode on Rx DMA, DMA_Cmd[1] — start/stop Tx DMA, DMA_Cmd[0] — start/stop Rx DMA, «1»-on, «0»-off	32'h00000000
0x11c	<i>reserved</i>			
-	<i>reserved</i>			
0x400..0x4c0	Stat_Monitor	R	См. Таблица 3 - Счетчики статистической информации	

<sup>1</sup> - Формат представления MAC адреса:

MAC\_tx\_addr\_10={byte1,byte0}, MAC\_tx\_addr\_32={byte3,byte2}, MAC\_tx\_addr\_54={byte5,byte4} — MAC адрес: «byte0-byte1-byte2-byte3-byte4-byte5»

<sup>2</sup> — Параметры имеют смысл при размере входного FIFO <1500 байт. Настоящий размер FIFO 2048 байт позволяет избежать разбиения принятых пакетов на части.

## 5. Счетчики статистической информации

Чтение счетчиков статистической информации выполняется через HOST интерфейс — Stat\_Monitor. Все счетчики несбрасываемые, циклические.

Таблица 3 - Счетчики статистической информации

Адрес	Счетчик	Описание	Примечание
0x00	Rx_Packet_Len_Ct	Суммарная длина всех принятых пакетов в байтах. Длина пакета определяется в учетом CRC (4 байта).	При приеме broadcast пакетов на первом отбрасываемом пакете будет дополнительно добавлено 4 байта из-за латентности схемы.
0x01	Rx_Packet_Ct	Общее количество принятых пакетов	У учетом отброшенных пакетов
0x02	Rx_Broadcast_Ct	Счетчик принятых broadcast пакетов. Отброшенные и неполные (количество байт в broadcast_bucket_depth меньше длины пришедшего пакета) broadcast пакеты не суммируются.	
0x03	Rx_Multicast_Ct	Счетчик принятых multicast пакетов	
0x04	Rx_Unicast_Ct	Счетчик принятых unicast пакетов	
0x05	Rx_CRC_Err_Ct	Счетчик CRC ошибок в принятых пакетах	
0x06	Rx_Fifo_Full_Err_Ct	Ошибки переполнения входного FIFO	
0x07	Rx_Too_Short_Too_Long_Ct	Счетчик принятых пакетов за пределами заданных длин пакетов	
0x08	Rx_Packet_Len<64	Счетчики количества принимаемых пакетов различной длины	
0x09	Rx_Packet_Len_64		
0x0a	Rx_Packet_Len_64_127		
0x0b	Rx_Packet_Len_128_255		
0x0c	Rx_Packet_Len_256_511		
0x0d	Rx_Packet_Len_512_1023		
0x0e	Rx_Packet_Len_1024_1517		
0x0f	Rx_Packet_Len>1518		
0x10	Rx_Pause_Frame_Ct	Счетчик принятых пакетов pause	
0x20	Tx_Packet_Len_Ct	Счетчик количества байт передаваемых пакетов	
0x21	Tx_Packet_Ct	Счетчик передаваемых пакетов	
0x22	Tx_Broadcast_Ct	Счетчик передаваемых broadcast пакетов	
0x23	Tx_Multicast_Ct	Счетчик передаваемых multicast пакетов	
0x24	Tx_Unicast_Ct	Счетчик передаваемых unicast пакетов	
0x25	Tx_Jam_Drop_Ct	Счетчик отброшенных пакетов при передаче <b>по завершению</b> счетчика повторов передачи при коллизиях в режиме Half Duplex	

0x26	Tx_Fifo_Underflow_Ct	Счетчик ошибок заполнения выходного FIFO	
0x27	Tx_Fifo_Overflow_Ct	Счетчик ошибок переполнения выходного FIFO	
0x28	Tx_Packet_Len<64	Счетчики количества передаваемых пакетов различной длины	
0x29	Tx_Packet_Len_64		
0x2a	Tx_Packet_Len_64_127		
0x2b	Tx_Packet_Len_128_255		
0x2c	Tx_Packet_Len_256_511		
0x2d	Tx_Packet_Len_512_1023		
0x2e	Tx_Packet_Len_1024_1517		
0x2f	Tx_Packet_Len>1518		
0x30	Tx_Pause_Frame_Ct		Счетчик передаваемых пакетов pause

## 6. DMA engine — описание работы, режимы

Программируемый DMA контроллер шины AXI4 с реализацией независимых каналов чтения и записи с поддержкой одиночного и блочного режимов передачи. Настройка контроллера выполняется через управляющие регистры.

Пары регистров:

**DMA\_Addr\_Start\_Tx**

**DMA\_Addr\_Finish\_Tx**

и

**DMA\_Addr\_Start\_Rx**

**DMA\_Addr\_Finish\_Rx**

Определяют границы кольцевых буферов чтения и записи соответственно. В регистры заносятся полные 32-разрядные адреса DMA шины.

Регистры настройки режимов работы DMA каналов чтения и записи:

**DMA\_Set\_Tx:**

DMA\_Set\_Tx[31] — одиночный/множественный режим («1» - множественный). Принудительное включение передачи бесконечного количества пакетов, независимо от значения в поле «количество пакетов»;

DMA\_Set\_Tx[30] — формирование прерывания после 1 пакета. Сигнал прерывания будет формироваться в конце обработки каждого пакета;

DMA\_Set\_Tx[29] — формирование прерывания после всех пакетов. Сигнал прерывания будет формироваться в конце передачи всех пакетов, количество которых указано в поле «количество пакетов». ВНИМАНИЕ! При установленном множественном режиме (бит 31 данного регистра) прерывание по этому признаку формироваться не будет, т.к. количество пакетов является бесконечным;

DMA\_Set\_Tx[28:12] - кол-во пакетов (до 128K), которое будет передано за один запуск работы блока;

DMA\_Set\_Tx[11:0] - размер пакета в байтах (до 1500байт). Важно отметить, что все пакеты при множественном режиме или значении в поле «количество пакетов» больше 1 имеют одинаковую длину. Для передачи нескольких пакетов с различной длиной требуется отдельно настраивать передачу каждого пакета с указанием его длины.

**DMA\_Set\_Rx**

DMA\_Set\_Rx[31] — одиночный/множественный режим («1» - множественный) Принудительное включение передачи бесконечного количества пакетов, независимо от значения в поле «количество пакетов»;

DMA\_Set\_Rx[30] — формирование прерывания после 1 пакета. Сигнал прерывания будет формироваться в конце обработки каждого пакета;

DMA\_Set\_Rx[29] — формирование прерывания после всех пакетов ВНИМАНИЕ! При установленном множественном режиме(бит 31 данного регистра) прерывание по этому признаку формироваться не будет, т.к. количество пакетов является бесконечным;

DMA\_Set\_Rx[28:12] - кол-во пакетов (до 128K), которое будет принято за один запуск работы блока;

DMA\_Set\_Rx[11:0] — резерв

Длина принятых пакетов подсчитывается автоматически и в приемный кольцевой буфер в системном ОЗУ принятые пакеты складываются в соответствии с их длиной, один за одним.

**DMA\_Cmd**

DMA\_Cmd[3] — burst mode on Tx DMA — разрешает блочный режим работы DMA канала чтения (AXI4 burst mode);

DMA\_Cmd[2] — burst mode on Rx DMA — разрешает блочный режим работы DMA канала записи (AXI4 burst mode);

DMA\_Cmd[1] — start/stop Tx DMA — запуск работы DMA канала чтения;

DMA\_Cmd[0] — start/stop Rx DMA — запуск работы DMA канала записи.

После установки бит DMA\_Cmd[1] и/или DMA\_Cmd[0] выполняется запуск DMA каналов и всего блока GMAC в соответствии с настройками в регистрах DMA\_Set\_Tx и DMA\_Set\_Rx.

## 7. Система синхронизации

GMAC блок использует несколько входных системных сигналов синхронизации:

DMA\_clk – синхросигнал DMA интерфейса,

HOST\_clk – синхросигнал HOST интерфейса, особых требований к сигналу не предъявляется;

CLK\_250 – опорный сигнал блока формирования сигналов синхронизации контроллера RX, TX MAC и MII/GMII интерфейсов.

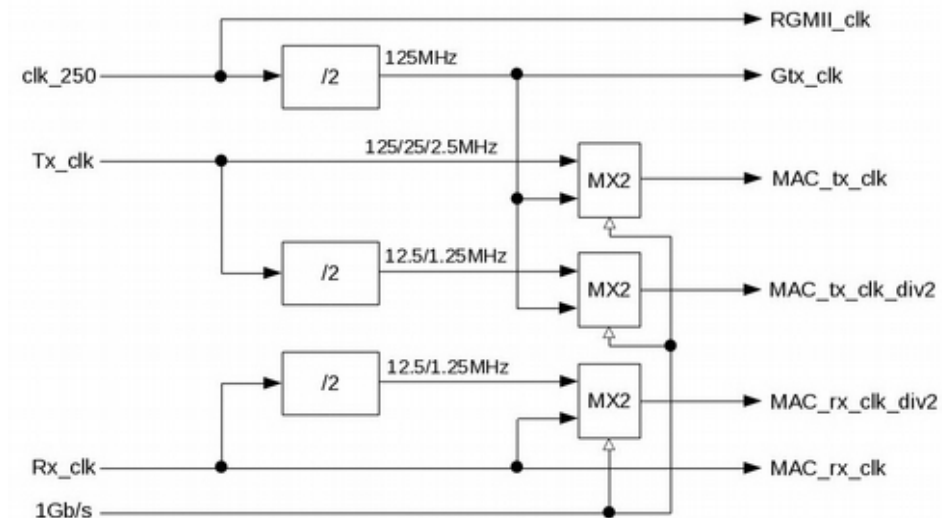


Рисунок 3 — Структура блока формирования сигналов синхронизации

## 8. Блоки памяти

В блоке используются блоки одно- и двухпортовой RAM (SPRAM, DPRAM). Характеристики блоков памяти зависят от целевой технологии, настроек и возможностей компилятора RAM.