# **Описание используемых типов**

## **Макросы**

|  |  |  |
| --- | --- | --- |
| Имя define | Значение | Описание |
| \_\_NOP | **asm**("nop;;") | Можно использовать в С-коде ассемблерную команду “nop” |
| \_\_RDS | **asm**("rds;;") | Можно использовать в С-коде ассемблерную команду “rds” |
| \_\_BTBEN | **asm**("btben;;") | Можно использовать в С-коде ассемблерную команду “btben” |

# **Описание используемых функций**

## **HAL\_SYS\_ExtBusEnable**

**Резюме**

**void** **HAL\_SYS\_ExtBusEnable**( SYS\_BusSize\_type size )

SYS\_BusSize\_type size – размер внешней шины в битах(*SYS\_BUS\_16b*, *SYS\_BUS\_32b*)

**Описание**

Функция включает внешнюю шину, путем установки в регистре PX\_ALT 0-6 битов для 32 битного режима или установки 0,1,4,5,6 битов и сброса 2 и 3 для 16 битного режима.

**Пример**

**#include** <hal\_1967VN044.h>

**int** **main**(**void**)

{

HAL\_SYS\_ExtBusEnable(*SYS\_BUS\_32b*);

}

## **HAL\_SYS\_ExtBusDisable**

**Резюме**

**void** **HAL\_SYS\_ExtBusDisable**( **void** )

**Описание**

Функция выключает внешнюю шину, путем сброса в регистре PX\_ALT 0-6 битов.

**Пример**

**#include** <hal\_1967VN044.h>

**int** **main**(**void**)

{

HAL\_SYS\_ExtBusDisable();

}

## **HAL\_SYS\_SDRAMEnable**

**Резюме**

**void** **HAL\_SYS\_SDRAMEnable**( uint32\_t sdrconValue )

uint32\_t sdrconValue – значение регистра SDRCON

**Описание**

Функция включает SDRAM, путем включения альтернативных функций (установка в единицу соответствующих бит регистра GPx\_ALT) в регистре GPC для 8-17 бит и записи sdrconValue в регистр SDRCON (для многократной записи в регистр SYSCON\SDRCON необходимо установить бит SYS\_WE в регистре CFG1). Функция завершается лишь после установки 13 бита регистра SYSTAT (т.е функция дожидается включения SDRAM). Так же для работы с внешней SDRAM необходимо включить внешнюю шину.

**Пример**

**#include** <hal\_1967VN044.h>

**int** **main**(**void**)

{

sdrconValue = SDRCON\_INIT | SDRCON\_RAS2PC5 | SDRCON\_PC2RAS2 |

SDRCON\_REF1100 | SDRCON\_PG1K | SDRCON\_CLAT2 | SDRCON\_ENBL;

HAL\_SYS\_ExtBusEnable(*SYS\_BUS\_32b*);

HAL\_SYS\_SDRAMEnable(sdrconValue);

}

## **HAL\_SYS\_SDRAMRegenPer**

**Резюме**

**void** **HAL\_SYS\_SDRAMRegenPer**( uint16\_t RegenPer )

uint16\_t RegenPer – значение периода регенерации SDRAM

**Описание**

Функция задает период регенерации SDRAM.

**Пример**

**#include** <hal\_1967VN044.h>

**int** **main**(**void**)

{

sdrconValue = SDRCON\_INIT | SDRCON\_RAS2PC5 | SDRCON\_PC2RAS2 |

SDRCON\_REF1100 | SDRCON\_PG1K | SDRCON\_CLAT2 | SDRCON\_ENBL;

HAL\_SYS\_ExtBusEnable(*SYS\_BUS\_32b*);

HAL\_SYS\_SDRAMRegenPer(16);

HAL\_SYS\_SDRAMEnable(sdrconValue);

}

## **HAL\_SYS\_SDRAMDisable**

**Резюме**

**void** **HAL\_SYS\_SDRAMDisable**( **void** )

**Описание**

Функция выключает SDRAM, путем отключения альтернативных функций (сброса в ноль соответствующих бит регистра GPx\_ALT) в регистре GPC для 8-17 бит и записи нуля в регистр SDRCON

**Пример**

**#include** <hal\_1967VN044.h>

**int** **main**(**void**)

{

HAL\_SYS\_SDRAMDisable();

}

## **HAL\_SYS\_SRAMEnable**

**Резюме**

**void** **HAL\_SYS\_SRAMEnable**( uint32\_t sysconValue )

uint32\_t sysconValue – значение регистра SYSCON

**Описание**

Функция включает SRAM, путем включения альтернативных функций (установка в единицу соответствующих бит регистра GPx\_ALT) в регистре GPC для 18-22 бит и записи sysconValue в регистр SYSCON (для многократной записи в регистр SYSCON\SDRCON необходимо установить бит SYS\_WE в регистре CFG1). Так же для работы с внешней SRAM необходимо включить внешнюю шину.

**Пример**

**#include** <hal\_1967VN044.h>

**int** **main**(**void**)

{

sysconValue = SYSCON\_MP\_WID64 | SYSCON\_MEM\_WID64 | SYSCON\_MSH\_PIPE2 |

SYSCON\_MSH\_WT0 | SYSCON\_MSH\_IDLE | SYSCON\_MS1\_PIPE1 |

SYSCON\_MS1\_WT0 | SYSCON\_MS1\_IDLE | SYSCON\_MS0\_SLOW |

SYSCON\_MS0\_WT3 | SYSCON\_MS0\_IDLE; HAL\_SYS\_ExtBusEnable(*SYS\_BUS\_32b*);

HAL\_SYS\_SRAMEnable(sysconValue);

}

## **HAL\_SYS\_SRAMDisable**

**Резюме**

**void** **HAL\_SYS\_SRAMDisable(** **void** )

**Описание**

Функция выключает SRAM, путем отключения альтернативных функций (сброса в ноль соответствующих бит регистра GPx\_ALT) в регистре GPC для 18-22 бит и записи нуля в регистр SYSCON

**Пример**

**#include** <hal\_1967VN044.h>

**int** **main**(**void**)

{

HAL\_SYS\_SDRAMDisable();

}

## **HAL\_SYS\_FlagEnable**

**Резюме**

**void** **HAL\_SYS\_FlagEnable**( **void** )

**Описание**

Функция включает Flag3-0, путем включения альтернативных функций (установка в единицу соответствующих бит регистра GPx\_ALT) в регистре GPC для 0-3 бит.

**Пример**

**#include** <hal\_1967VN044.h>

**int** **main**(**void**)

{

HAL\_SYS\_FlagEnable();

}

## **HAL\_SYS\_FlagDisable**

**Резюме**

**void** **HAL\_SYS\_FlagDisable**( **void** )

**Описание**

Функция выключает FLAG3-0, путем отключения альтернативных функций (сброса в ноль соответствующих бит регистра GPx\_ALT) в регистре GPC для 0-3 бит.

**Пример**

**#include** <hal\_1967VN044.h>

**int** **main**(**void**)

{

HAL\_SYS\_FlagDisable();

}

## **HAL\_SYS\_CorePrimariPriority**

**Резюме**

**void** **HAL\_SYS\_CorePrimariPriority**( **void** )

**Описание**

Функция устанавливает приоритет процессора над DMA при доступе к внешней памяти путем сброса 12 бита в регистре CFG1. Есть так же обратная функция HAL\_DMA\_PrimariPriority().

**Пример**

**#include** <hal\_1967VN044.h>

**int** **main**(**void**)

{

HAL\_SYS\_CorePrimariPriority();

}

## **HAL\_SYS\_WaitMs**

**Резюме**

**void** **HAL\_SYS\_WaitMs**( uint32\_t ms )

uint32\_t ms – количество миллисекунд

**Описание**

Функция задержки на заданное количество миллисекунд. Для своей работы использует значение частоты тактирования ядра, хранящееся в структуре PLL\_Freq.

**Пример**

**#include** <hal\_1967VN044.h>

**int** **main**(**void**)

{

HAL\_SYS\_WaitMs(100);

}

## **HAL\_SYS\_WaitClk**

**Резюме**

**static inline void** **HAL\_SYS\_WaitClk**( uint32\_t clk )

uint32\_t clk – количество тактов

**Описание**

Функция задержки на заданное количество тактов.

**Пример**

**#include** <hal\_1967VN044.h>

**int** **main**(**void**)

{

HAL\_SYS\_WaitClk(500);

}