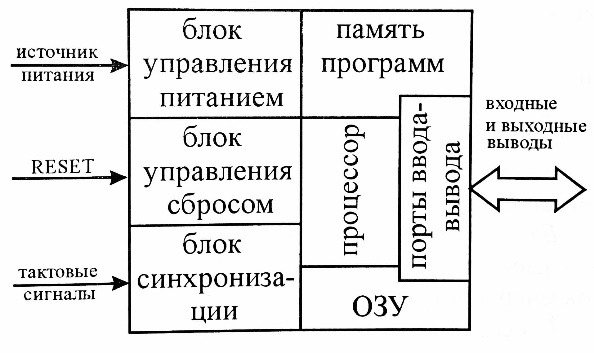
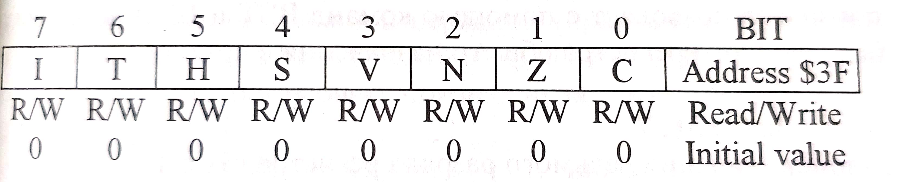
**Вопросы по микроконтроллерам**

**Весна 2021 года**

1. **Ваша фамилия, имя, отчество, номер группы.**  
   Кошелев Сергей Маркович Б01-908  
   Закирьянов Равиль Венерович Б01-908  
   Титов Егор Михайлович Б01-909
2. **Фамилия, имя, отчество лектора.**Донов Геннадий Иннокентьевич
3. **Чем отличается микроконтроллер от микропроцессора.**Микроконтроллер имеет более сложную структуру, у него есть порты ввода-вывода, ОЗУ, память программ, таймеры и т.д. Микропроцессор – это лишь исполняющее ядро
4. **Какие тактовые частоты могут быть у ATmega8535.**0-16 МГц(36Мгц — разгон (форум) 65Мгц (atmega328, жидкий азот))
5. **Какие таймеры есть у ATmega8535.**Таймер 0 (8bit), Таймер 1(16bit), Таймер 2(8bit)
6. **Внутренняя структура МК.**Процессор – обработка информации, исполнение команд  
   Память программ – хранит программу, в соответствии с которой работает МК  
   ОЗУ (оперативное запоминающее устройство) – хранение промежуточных результатов (RAM)  
   Порты ввода/вывода – обмен информацией с внешнем миром  
   Блок управления питанием – правильный запуск при включении питания  
   Блок управления сбросом – установка в исходное состояние вместе с входом RESET  
   Блок синхронизации – выработка тактовых сигналов, необходимых для правильного взаимодействия всех внутренних блоков МК
7. **Какие значения записаны в TCCR после сигнала RESET.**Все биты станут нулями
8. **Порт А. Сколько прерываний и сколько регистров ввода/вывода принадлежит порту А. Назначение этих регистров ввода/вывода.**PORTA – регистр данных порта А  
   DDRA – регистр выбора направления передачи данных порта А  
   PINA – нельзя ничего записать, при чтении из него будет прочитано то, что в данный момент присутствует на выводах порта А  
   Прерываний у порта А нет
9. **Регистр SREG. Назначение его разрядов.**SREG (status register) – 8-разрядный регистр, регистр флагов.  
   Бит 0 – С: признак переноса  
   Бит 1 – Z: признак нуля  
   Бит 2 – N: признак отрицательного результата  
   Бит 3 – V: признак переполнения  
   Бит 4 – S: равен сумме по модулю 2 содержимого третьего и второго разрядов  
   Бит 5 – H: признак переноса между полубайтами  
   Бит 6 – T: временное хранение бита  
   Бит 7 – I: глобальное разрешение прерывания
10. **Почему после сигнала RESET все прерывания запрещены.**Для обеспечения корректной работы МК
11. **Приведите пример использования разряда Т в регистре SREG.**Передача битов из одного регистра общего назначения в другой:

**bst r31, 7;** запись значения седьмого разряда регистра r31 в T

**bld r0, 3;** запись из Т в третий разряд регистра r0

1. **Таймер 0. Режимы работы, количество прерываний, регистры ввода/вывода, принадлежащие таймеру 0.**Режимы работы:  
   Normal (режим 0) – обычный суммирующий счетчик;  
   Phase Correct PWM (режим 1) – ШИМ с точной фазой;  
   CTC (режим 2) – счет по модулю (регистр OCR0);  
   Fast PWM (режим 3) – быстродействующий ШИМ.  
   Прерывания 2: TIMER0\_OVF (переполнение) и TIMER0\_COMP (порог).  
   Регистры ввода/вывода: TCNT0, TCCR0, SFIOR, TIMSK (совместно с таймером 1 и таймером 2), TIFR (совместно с таймером 1 и таймером 2)
2. **В каких режимах таймера 0 порог изменяется не сразу (двойная буферизация записи) при записи нового значения в регистр порога с помощью команды OUT.**Режимы ШИМ (1 и 3)
3. **Откуда приходит сигнал на вход TCNTO.**С выхода управляемого предварительного делителя частоты (prescaler).
4. **Как можно разрешить (запретить) прерывания по переполнению таймера 0.**С помощью 7 разряда регистра флагов SREG, разрешающего общие прерывания и разряда 0 регистра флагов TIMSK (разрешено – обе единицы, запрещено – хотя бы один ноль).
5. **Написать программу с использованием таймера 0, вырабатывающую симметричное прямоугольное колебание на одном из выходов порта А.**#include <avr/io.h>

#include <avr/interrupt.h>

.global TIMER0\_COMP

TIMER0\_COMP:

in r16, PORTA

eor r16, r17

out PORTA, r16

reti

.global main

main:

sbi DDRA, DDA0 ;set PA0 as OUTPUT

cbi PORTA, PORTA0 ;pull PA0 to grownd

ldi r17, 1 << PORTA0

sei ;enable interrupts

ldi r16, 1 << OCIE0 ;enable comparison interrupt

out TIMSK, r16

ldi r16, 0x7f ;treshold on 50%

out OCR0, r16

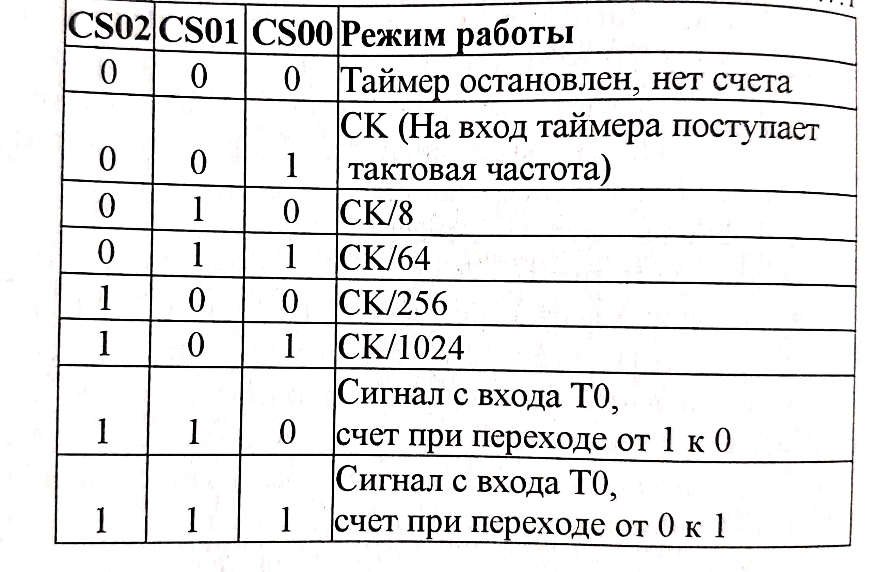
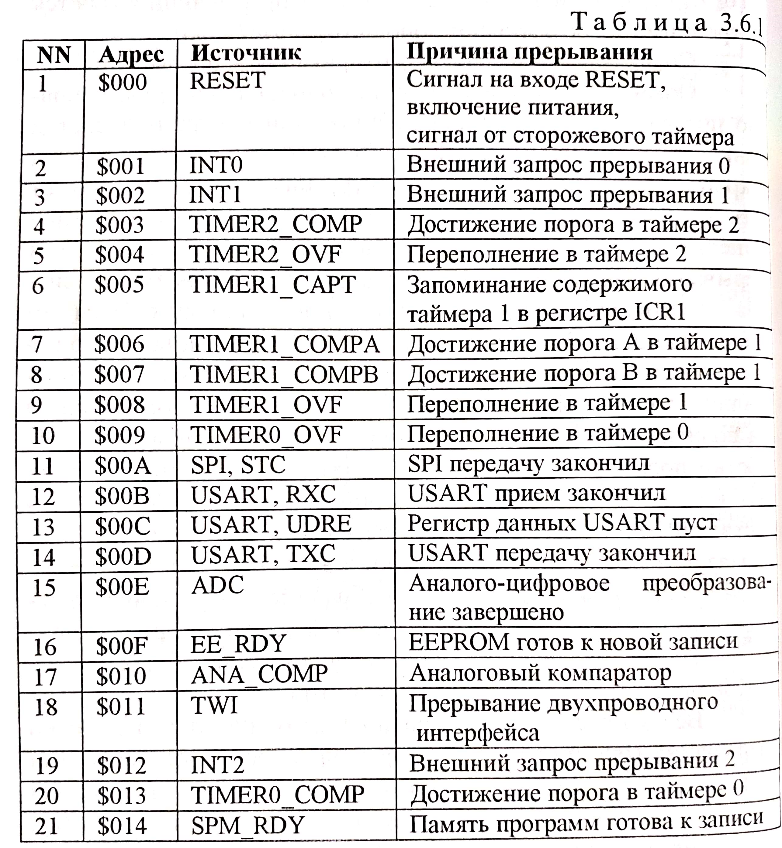
ldi r16, 1 << WGM00 | 1 << CS00 ;phase-correct PWM ;no prescaling

out TCCR0, r16

loop:

nop

rjmp loop

1. **Какие коэффициенты деления частоты позволяет получать предварительный делитель таймера 0.**8, 64, 256, 1024
2. **Какой режим таймера 0 позволяет вырабатывать треугольные колебания, используя дополнительную интегрирующую цепочку.**Для треугольных колебаний необходимо прямоугольные перед интегрирующей цепочкой, поэтому подойдет любой режим
3. **Как запрограммировать предварительный делитель таймера 0.**Используются три разряда регистра TCCR0 (2, 1, 0). Значения разрядов программируют делитель согласно таблице:  
   
4. **Режим 0 таймера 0.**Normal. TCNT0 – обычный суммирующий счетчик. По каждому импульсу тактового сигнала, поступающего с выхода предварительного делителя, содержимое TCNT0 увеличивается на единицу. При переполнении – прерывание по переполнению, счет продолжается с $00. Если достигнуто пороговое значение – прерывание по сравнению.
5. **Режим 1 таймера 0.**Phase Correct PWM, ШИМ с точной фазой. Генерация сигналов с широтно-импульсной модуляцией. TCNT0 как реверсивный счетчик, изменение состояния по каждому импульсу такового сигнала, поступающего от предварительного делителя. Состояние счетчика сначала увеличивается до максимума, потом уменьшается до нуля. Переполнение по возвращению в 0, изменение выхода по достижению порога.
6. **Режим 2 таймера 0.**CTC, счет по модулю. Ообнуление TCNT0 после того как содержимое сравняется с содержимым регистра OCR0. Прерывание по переполнению при достижении верхней границы.
7. **Режим 3 таймера 0.**Fast PWM, быстрый ШИМ. Позволяет генерировать высокочастотный сигнал с широтно-импульсной модуляцией. Изменение от нуля до $FF, потом прерывание по переполнению и снова обнуление. Доступно прерывания при достижении порога.
8. **Когда меняется порог в режиме 3 таймера 0.**Особенность этого режима, что записываемое число сохраняется в специальном буферном регистре. Изменение содержимого порога происходит только после достижения счетчиком максимума $FF.
9. **Можно ли писать в TCNT0 без остановки счета.**Можно, разработчики предприняли меры для записи и чтения без остановки
10. **Как можно остановить счет в таймере 0.**Записать нули в соответствующие разряды регистра TCCR0 (см таблицу из 19 вопроса).
11. **Система прерываний микроконтроллера ATmega8535.**Обнаружив запрос, система прерываний приостанавливает работы основной программы и запускает некоторую другую программу, которую называют программу обработки прерываний. Для каждого прерывания должна быть написана своя программа обработки. Закончив свою работу, программа обработки прерывания должна обеспечить возвращение к прерванной программе  
    Запросы поступают на блок обработки; определяется его номер; проверяется, разрешено ли прерывание. Если разрешено, блокируются остальные, содержимое программного счетчика заносится в стек.
12. **Сколько всего прерываний у ATmega8535.**21: 4 из которых являются внешними, 7 – внутренними
13. **Как организовать вложенные прерывания.**Нужно разрешить (запись единицы в 7 разряд регистра флагов) глобальные прерывания при обработке прерывания.
14. **Как можно разрешить (запретить) одновременно все прерывания.**cli – общее запрещение  
    sei – общее разрешение
15. **Как организована система приоритетов при обработке прерываний.**Существует таблица прерываний:  
      
    Первым обрабатывается прерывание, имеющее наименьший номер.
16. **Какое минимальное время требуется для преобразования в АЦП.**65 мкс
17. **Чем сигнальный процессор отличается от МК.**У него есть специфичный набор команд и регистров, предназначенных для уменьшения времени обработки сигналов. Основная задача МК – работа с периферией.
18. **Зачем в программе надо устанавливать начальное значение Stack Pointer и чему это значение должно быть равно.**После RESET правильное значение не гарантировано, поэтому надо ставить его вручную. Выставляется на конец памяти.
19. **Сторожевой таймер и особенности его работы.**WatchDog Timer. Предназначен для ликвидации последствий сбоев в работе МК, возникающих из-за различного рода помех.  
    Если он включен, то через некоторых промежуток времени он вырабатывает сигнал сброса RESET, перезапуская рабочую программу.
20. **Что такое SPI и зачем он нужен.**Последовательный синхронный интерфейс. Позволяет с высокой скоростью передавать данные между МК Atmega8535 и различными внешними устройствами.
21. **Как инициировать передачу байта в SPI.**Запись байта в SPDR
22. **Сколько прерываний и сколько регистров ввода/вывода принадлежит SPI.**Регистры:  
    SPDR (Data Register), SPCR (Control Register), SPSR (Status Register)  
    1 прерывание: после передачи каждого байта
23. **Далее пойдут вопросы про однопроводный интерфейс (сеть MicroLAN).**Ок!
24. **Сколько проводов необходимо для реализации однопроводного интерфейса.**1(только данные) 2 (данные + земля) 3 (как у нас: данные земля и питание). Зависит от устройства схемы.
25. **Как выглядит физический ноль и физическая единица.**1: напряжение выше порогового  
    0: напряжение ниже порогового
26. **Как в однопроводном интерфейсе передается информационный ноль и информационная единица? Какова максимальная скорость передачи?**Ноль – физический ноль 60 мкс (длительный импульс в начале тайм слота, потом возврат в 1)   
    Единица – физический ноль не более 15 мкс (короткий импульс в начале тайм слота, потом возврат в 1)Очередная передача не раньше чем через 60 микросекунд после начала предыдущей (минимальный размер тайм слота)
27. **Что такое серийный номер в однопроводном интерфейсе и какова его структура.**Первые 8 бит - код семейства, 48 бит - серийный номер, Последние 8 бит - контрольная сумма CRC8.
28. **Какая команда позволяет Master определить номера всех Slave в сети MicroLAN.**

SEARCH\_ROM (0xF0) или частный случай ALARM SEARCH (0xEC).

1. **Как выглядит сигнал сброса в сети MicroLAN.**

Физический ноль на минимум 480мкс. Затем Физическая единица на 15-60мкс.