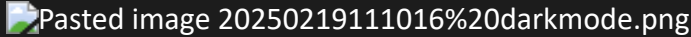


## Процесс выполнения команд AVR

1 командный цикл - 2 машинных цикла

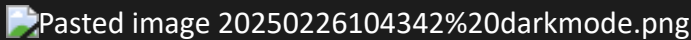
1 машинный цикл - 1 машинный такт



## Программно доступные ресурсы ATmega8515

- **32 регистра общего назначения R0-31**, 8-разрядные, в т.ч. X-Z сдвоенные
- **64 регистра ввода-вывода**, в т.ч.:
  - Программный счётчик PC (Program Counter), 16-разрядный
  - Указатель стека SP (Stack Pointer), 16-разрядный
  - Регистр состояния процессора (SREG), 8-разрядный
  - Регистры управления MCUCR, EMCUCR
  - Регистры периферии:
    - Порты в/в (DDRx, PORTx, PINx)
    - Таймеров
    - Системы прерываний (GICR, GIFR)
    - Контроллеров интерфейсов SPI, UART
    - ... (и тд походу)
- **Ячейки памяти**
  - 512 байт внутренней памяти RAM
  - 512 байт внутренней энергонезависимой памяти EEPROM
  - Ячейки внешней памяти данных
  - Ячейки Flash-памяти

## Регистр состояния процессора SREG



I - Interrupt Enable - глобальное разрешение прерываний

T - Transfer Bit - бит для временного хранения данных

C - Carry Flag - флаг переноса

H - Half Carry Flag - флаг полупереноса (переноса между тетрадами)

Z - Zero Flag - флаг нуля, =1, если результат последнего действия =0

N - Negative Flag - флаг минуса

V - Two's Complement Overflow Flag (where the fuck did V come from) - флаг переполнения допкода (вспоминай информатику 1 семестр) - равен XOR между двумя последними переносами при действии  $V = p8 \oplus p7$

S - Sign Flag - флаг знака, не зависит от переполнения  $S = N \oplus V$

P.S. Как работают знаковые числа в проце: первый бит - знак (1 если минус), и если число отрицательно, то все 8 бит рассматриваются как допкод, то есть чтобы получить десятичное число из битовой записи:

$A = 1010.1010$  код -  $01010110$  (!A + 1),  $01010110 = 86 \rightarrow A = 1010.1010 = -86$

## Архитектуры системы команд

- **CISC** - Complex Instruction Set Computer
  - Сложные команды, выполняющие сразу несколько операций
  - Команды разной длины
  - Проще прога на ассемблере, итоговая программа компактнее
- **RISC** - Reduced Instruction Set Computer
  - Простые команды, каждая выполняет отдельную операцию
  - Команды одинаковой длины
  - Проще аппаратная реализация процессора, в т.ч конвейера
  - Проще оптимизация кода на языках высокого уровня

Пример - сложение регистра с числом по адресу

- CISC (MSC-51):  
ADD A, 80H
- RISC (AVR):  
LDR R1, 0x80  
ADD R0, R1