

# Ассемблер Atmel AVR

Занятие №4: Регистр статуса, стек,  
подпрограммы

# Память AVR

Flash 16-bits

0x0000

Память программ

FLASHEND – 0xFFFF

SRAM 8-bits

0x0000 POH 0x001F

0x0020 I/O 0x005F

0x0060

Внутренняя SRAM

RAMEND

RAMEND+1

Внешняя SRAM

0xFFFF

EEPROM 8-bits

0x0000

Память EEPROM

EEPROMEND – 0xFFFF

# Стек



**Стек** (англ. Stack – стопка) — структура данных, представляющая собой список элементов, организованных по принципу LIFO.

**LIFO** (англ. last in first out) — принцип когда элемент пришедший последним выходит первым.

# Стек в AVR



**SP** (Stack Pointer) – указатель стека, указывает на начало стека. Стек растет в верх, указатель стека вниз. При добавлении элемента указатель стека уменьшается

**SPH, SPL** – Если у МК больше 256 байт памяти для адресации стека требуется 16 бит.

# OUT – Записать данные из регистра в порт I/O

**Синтаксис:** OUT Rr,

**Размер:** 2

Rr

байта

1011	1PPr	rrrr	PPPP
------	------	------	------

**Операнды:**  $0 \leq r \leq 31$ ,  $0 \leq P \leq 63$

**Счетчик:** PC +=

**Циклы:**

1	I	T	H	S	V	N <sup>1</sup>	Z	C
–	–	–	–	–	–	–	–	–

**Определение:** Команда сохраняет данные регистра Rr в пространстве I/O памяти SRAM

# Инициализация стека

```
LDI R16, Low (RAMEND)
```

```
OUT SPL, R16
```

```
LDI R16, High (RAMEND)
```

```
OUT SPH, R16
```

# PUSH – Загрузить регистр в стек

**Синтаксис:** PUSH

**Размер:** 2

Rd

байта

1001	001d	dddd	1111
------	------	------	------

**Операнды:**  $0 \leq d \leq 31$

**Счетчик:** PC +=

**Циклы:**

1	I	T	H	S	V	N <sup>2</sup>	Z	C
–	–	–	–	–	–	–	–	–

**Определение:** Команда помещает содержимое регистра Rd в стек.

# POP – Загрузить значение из стека в регистр

**Синтаксис:** POP

**Размер:** 2

Rd

байта

1001	000d	dddd	1111
------	------	------	------

**Операнды:**  $0 \leq d \leq 31$

**Счетчик:** PC +=

**Циклы:**

1	I	T	H	S	V	N <sup>2</sup>	Z	C
–	–	–	–	–	–	–	–	–

**Определение:** Команда загружает регистр Rd байтом содержимого стека.



# Что таится за этими командами

на самом деле:

PUSH R16

SRAM[SP] = R16

SP--

POP R16

SP++

R16 = SRAM[SP]

# IN – Загрузить данные из порта I/O в регистр

**Синтаксис:** IN Rd,

**Размер:** 2

P

байта

1011	0PPd	dddd	PPPP
------	------	------	------

**Операнды:**  $0 \leq d \leq 31$ ,  $0 \leq P \leq 63$

**Счетчик:** PC +=

**Циклы:**

1	I	T	H	S	V	N <sup>1</sup>	Z	C
–	–	–	–	–	–	–	–	–

**Определение:** Команда загружает данные из пространства I/O памяти SRAM в регистр Rd

# Пример того как реализовывается 2й стек

```
LDI R16, 24  
PUSH R16  
LDI R16, 32  
PUSH R16  
LDI R16, 65  
PUSH R16
```

```
IN R30, SPH  
IN R31, SPL
```

```
LDI R16, low(RAMEND-100)  
OUT SPL, R16  
LDI R16, high(RAMEND-100)  
OUT SPH, R16
```

```
LDI R16, 68  
PUSH R16  
LDI R16, 12  
PUSH R16  
LDI R16, 5  
PUSH R16
```

```
IN R28, SPH  
IN R29, SPL
```

```
OUT SPL, R31  
OUT SPH, R30
```

```
POP R17
```