CPU Registers

1: General Purpose Registers
2: Status Register – contains info about result of last executed instruction
3: Program Counter – Address of next instruction
4: Instruction Register – Holds fetched instruction
5: Stack Pointer – Points to top of stack
Stack

- Used for storing temporary data
  - Local variables
  - Return addresses after interrupts or subroutine
- Implemented as growing from higher to lower address
  - Initial pointer set equal to last address of SRAM
- Push – decreases SP
- Pop – increases SP
Stack Pointer - SP

- Points to top of the stack
  - Implemented as two 8-bit registers
- What does address in SP tell us?

SP implementation in AVR

<table>
<thead>
<tr>
<th>Bit</th>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x3E (0x5E)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>SP10</td>
<td>SP9</td>
</tr>
<tr>
<td>0x3D (0x5D)</td>
<td>SP7</td>
<td>SP6</td>
<td>SP5</td>
<td>SP4</td>
<td>SP3</td>
<td>SP2</td>
<td>SP1</td>
<td>SP0</td>
</tr>
<tr>
<td>Read/Write</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
</tr>
<tr>
<td>Initial Value</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SPH</td>
<td>SPL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Status Register (SREG)

- Contains information about the result of the most recently executed instruction
  - Use to alter program flow on conditional operations
- NOT automatically stored when entering an interrupt routine
  - Must be handled by software
Status Register (SREG)

I: Global Interrupt Enable – Allows all interrupts
T: Bit Copy Storage – Used with BLD and BST for loading and storing bits from one register to another
H: Half Carry Flag – Indicates half-carry in some arithmetic
S: Sign flag – always XOR of V and N
V: Two’s complement overflow flag
N: Negative Flag
Z: Zero Flag
C: Carry Flag – Indicates carry in arithmetic operation
Register File

- Most instructions have direct access to all registers
  - LD, MUL
- Some only operate on R16-R31
  - ANDI, CPI, SUBI, MULS
- A few operate only on R16:R23
  - Special multiply operations
- Double word operations operate on register pairs (R24-31)
- Most single-byte register or register+immediate operations are single cycle instructions
Special Purpose Registers - X, Y, Z

- Registers R26-R31 have a special purpose
  - In addition to allowing general purpose usage
- 16-bit address pointers for addressing data space
- Used in functions for different addressing modes
  - Fixed displacement
  - Automatic increment/decrement

![Figure 6-5. The X-, Y-, and Z-registers](image-url)