

Prelab 4:

1. What is the stack pointer? How is the stack pointer used, and how do you initialize it? Provide pseudocode (not actual assembly code) that illustrates how to initialize the stack pointer.

The stack pointer is a register that always points to the top of the Stack, where temporary data, local variables, and return addresses are stored. The AVR stack pointer consists of 2 8 bit registers: stack pointer high (SPH) and stack pointer low (SPL). To initialize the stack pointer, both ends (high and low) will need to be stored with the initial address to point to.

Pseudocode:

Load lower byte of SRAM address into the lower stack pointer register, then load the higher byte of SRAM address into the high stack pointer register.

2. What does the AVR instruction LPM do, and how do you use it? Provide pseudocode (not actual assembly code) that shows how to setup and use the LPM instruction.

LPM - Load Program Memory loads one byte from the Z register, into a destination register.

An example in pseudocode:

First initialize a Z-Pointer (high and low), and use LPM to store it into a general purpose register.

3. Take a look at the definition file m128def.inc (This file can be found in the Solution Explorer → Dependencies folder in Atmel Studio, assuming you have an Assembler project open and you have already built an assembly program that includes this definition file. Two good examples of such a project would be your Lab 1 and Lab 3 projects.) What is contained within this definition file? What are some of the benefits of using a definition file like this? Please be specific, and give a couple examples if possible.

The title of the file is register/bit definitions for the ATmega128. This file maps many different definitions with names for the device ATmega128. One very useful benefit of having this file is the ability to use the X, Y, & Z registers as they are defined here. Status registers are also initialized here and those are used in many different logical processes.