Lab 5 Memories and Index Registers

Submit lab5.asm by 5:00pm on February 9th, 2018.

I. AVR ATmega2560 Memories

Program storage: Flash memory (.cseg directive)
Data storage: SRAM and EEPRROM (.dseg and .eseg directives)

The Configuration Summary of ATmega2560:[1]

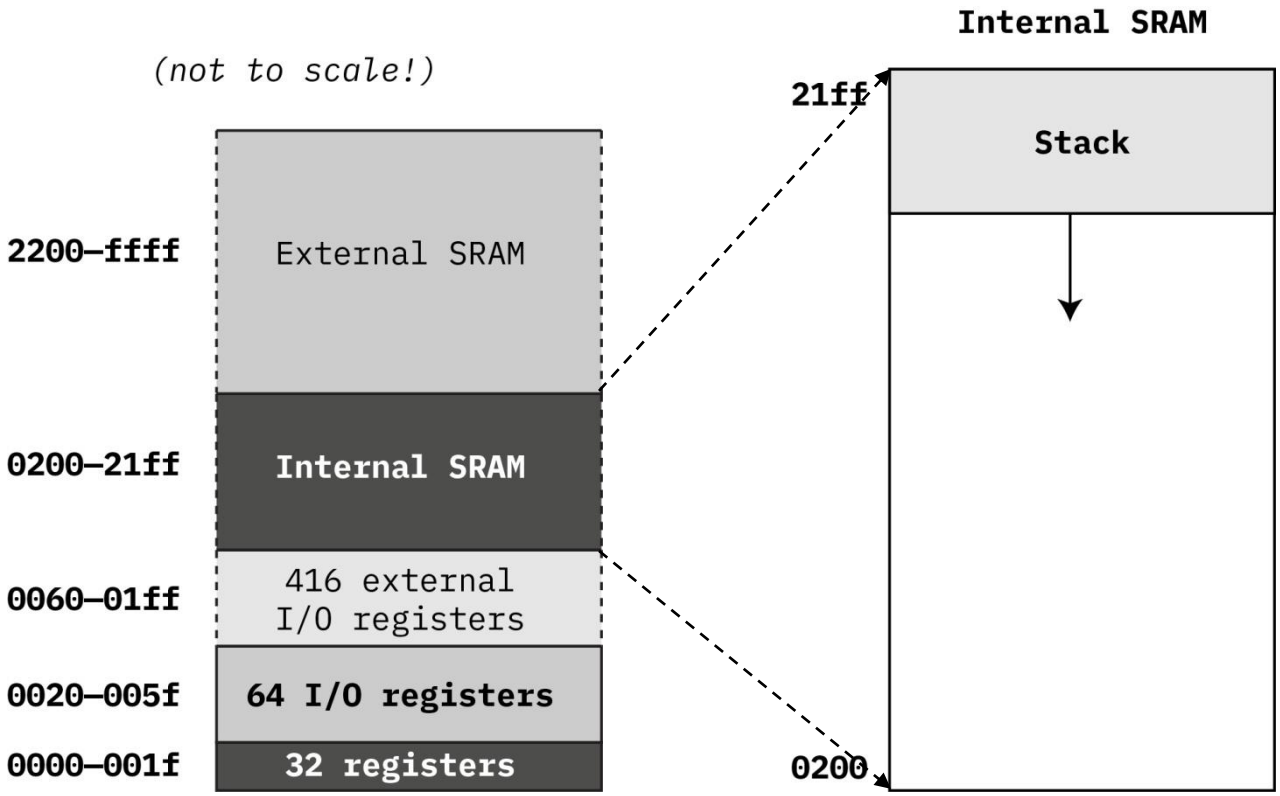
Table 2-1. Configuration Summary

Device	Flash	EEPROM	RAM	General Purpose I/O pins	16 bits resolution PWM channels	Serial USARTs	ADC Channels
ATmega2560	256KB	4KB	8KB	86	12	4	16

What is the largest flash memory address of ATmega2560?

Based on Table 2-1, the memory size is 256KB -> 2^{18}Bytes -> 2^{17}words, the largest word address is 0x1FFFF(17bits).

The diagram of the data memory (SRAM): [2]



II. Index Registers

Data Direct Addressing:

We practiced the following instructions in the assignment 1:

```
lds R16, IN1 ;(reverse.asm)
sts PARITY, R20  ;(in parity.asm)
```

In general, the instruction takes the form of "lds Rd, (k)", where the value of k is a 16-bit unsigned integer representing memory address of the SRAM (data memory). The content (1 byte) stored in memory address k is loaded to register Rd.

Data Indirect Addressing:

The AVR processor has three register pairs that can be used for data indirect addressing. The three register pairs (also called index registers) are:

```
X -> R27:R26 or XH :XL
Y -> R29:R28 or YH :YL
Z -> R31:R30 or ZH :ZL
```

The address to be accessed must be preloaded into either X, Y, or Z register. What do the following statements do?

```
LD Rd, X
ST X, Rr
LPM R23, Z+
```

Indirect addressing is especially suited for accessing arrays, tables, and Stack Pointer.

III. Download lab5.asm and finish the program.

A C-style string (C string - the last byte of the string is 0x00) is stored in the program memory (flash memory). Write a short program to calculate the length of the string and store the length to the data memory (SRAM). To be specific, the subroutine/function will be labelled "str_length", that the main assembly code will call this routine to compute the string length, and finally it is the main routine that saves the length in data memory. The suggested structure of your program is:

```
------------------- main program -----------------
   Get the memory address of msg1 to srcH:srcL – the register pair to store string address
   Call subroutine str_length and store the returned value in register named n
   Save n to data memory location named LENGTH1

   Get the memory address of msg2 to srcH:srcL – the register pair to store string address
   Call subroutine str_length and store the returned value in register named n
   Save n to data memory location named LENGTH2

   Get the memory address of msg3 to srcH:srcL – the register pair to store string address
   Call subroutine str_length and store the returned value in register named n
```

Save n to data memory location named LENGTH3

Initialize msg1 in the program memory
Initialize msg2 in the program memory
Initialize msg3 in the program memory
---------- end of the main program ----------

---------- subroutine str_length ----------
Copy srcH to ZH, the high byte of the index register Z
Copy srcL to ZL, the low byte of the index register Z
Initialize a register (counter) to store the number of characters
Write a loop, load one character at a time from program memory
    Increment counter
    If the character is 0, end of the loop
    Else, back to the loop
---------- end of subroutine str_length ----------

Submit lab5.asm by 5:00pm on February 9th, 2018.

This lab is derived from Chapters 5 and 12 of your textbook (Some Assembly Required by Timothy S. Margush) and the datasheet of ATMEGA 2560.

1. The diagram is copied and modified from the datasheet mentioned above.
2. The diagrams are provided by Dr. Mike Zastre.