

LAB 3-A

STACK MEMORY

OBJECTIVES:

- ☐ To examine the stack.

MATERIAL:

- ☐ Atmel Studio.

WEB SITES:

- ☐ www.microchip.com for Atmel Studio Software

ACTIVITY 1

Write and assemble a program to load values \$20, \$31, \$42, \$53, and \$64 into each of registers R20 to R24 and then push each of these registers onto the stack. Single-step the program, and examine the stack and the SP register after the execution of each instruction.

```
.INCLUDE "m328pdef.inc"
```

```
ORG 0x0000
```

```
LDI R16, LOW(RAMEND)
```

```
OUT SPL, R16
```

```
LDI R16, HIGH(RAMEND)
```

```
OUT SPH, R16
```

```
LDI R20, $20
```

```
LDI R21, $31
```

```
LDI R22, $42
```

```
LDI R23, $53
```

```
LDI R24, $64
```

```
PUSH R20
```

```
PUSH R21
```

LAB 3-A

STACK MEMORY

PUSH R22

PUSH R23

PUSH R24

HERE: RJMP HERE

ACTIVITY 2

Write and assemble a program to:

- Set SP = \$29D, -
- Store (without using push operation) a different value 6, 5, 4, 3, 2, 1 in RAM locations \$29D, \$29C, \$29B, \$29A, \$299, and \$298, respectively
- POP each stack location into registers R20 – R24.
- Use the simulator to single-step and examine the registers, the stack, and the stack pointer.

```
INCLUDE "m328pdef.inc"
```

```
.ORG 0x00
```

```
; 1. Set SP = $29D
```

```
LDI R16, LOW(0x29D)
```

```
OUT SPL, R16
```

```
LDI R16, HIGH(0x29D)
```

```
OUT SPH, R16
```

```
; 2. Store values 6,5,4,3,2,1 into RAM $29D down to $298
```

```
; We use STS (Store to SRAM) to bypass the PUSH instruction
```

```
LDI R16, 6
```

```
STS 0x29D, R16
```

```
LDI R16, 5
```

```
STS 0x29C, R16
```

LAB 3-A

STACK MEMORY

LDI R16, 4

STS 0x29B, R16

LDI R16, 3

STS 0x29A, R16

LDI R16, 2

STS 0x299, R16

LDI R16, 1

STS 0x298, R16

; 3. POP each location into R20 - R25 (Adjusting SP as we go)

; Note: POP increments the SP *after* loading the value.

; To POP from \$298, the SP must be at \$298.

LDI R16, LOW(0x298)

OUT SPL, R16

LDI R16, HIGH(0x298)

OUT SPH, R16

POP R20 ; Loads value from \$298 (1), SP becomes \$299

POP R21 ; Loads value from \$299 (2), SP becomes \$29A

POP R22 ; Loads value from \$29A (3), SP becomes \$29B

POP R23 ; Loads value from \$29B (4), SP becomes \$29C

POP R24 ; Loads value from \$29C (5), SP becomes \$29D

POP R25 ; Loads value from \$29D (6), SP becomes \$29E

HERE: RJMP HERE

LAB 3-A

STACK MEMORY

From Activity 1 and 2, answer the following questions:

- 1) Upon reset, what is the value in the SP register?

On reset, the SP register will be 0x0000 but cannot be used for the stack.

RAMEND (For the ATmega328P, this is typically addressed at \$08FF).

- 2) Upon pushing data onto the stack, the SP register is decremented
(decremented, incremented).

- 3) Upon popping data from the stack, the SP register is incremented
(decremented, incremented).

- 4) Can you change the value of the SP register? If yes, explain why you would want to do that.

Yes, we can change the value of the SP register. We can modify the Stack Pointer (SP) by using the OUT instruction to write specific addresses into the SPL and SPH I/O registers. The reason to change is if different tasks need different stacks, handle overflow, adjust stack/heap.