

PRINCIPLES AND APPLICATION OF MICROCONTROLLERS

AVR Assembly Lab10: Assembly Program Simulation

Deliverables

Create an assembly project, type the program in Fig. 9, build it, and answer the questions.

```
.EQU SUM = 0x300
LDI R16, 0x25
LDI R17, 0x34
LDI R18, 0x31
ADD R16, R17
ADD R16, R18
LDI R17, 0x74
ADD R16, R17
LDI R20, 0
ADD R16, R20
LDI R21, 0xFF
OUT DDRD, R21
L2: INC R20
OUT PORTD, R20
STS SUM, R20
RJMP L2
```

Figure 9: Deliverable program code

1. Use the Atmel Studio debugging functions to find the machine codes for the program in Fig. 9. Make a copy of the machine codes to your report.

```
000000 e205
000001 e314
000002 e321
000003 0f01
000004 0f02
000005 e714
000006 0f01
000007 e040
000008 0f04
000009 ef5f
00000a b95a
00000b 9543
00000c b94b
00000d 9340 0300
```

2. Identify the opcodes for the instructions in the following table. Report them in binary forms. Leave the digit cross (X) if it is a part of an operand.

Instruction	Opcode
LDI	1110 XXXX XXXX XXXX
ADD	0000 11XX XXXX XXXX
OUT	1011 1XXX XXXX XXXX

INC	1011 010X XXXX 0011
STS	1001 001X XXXX 0000 XXXX XXXX XXXX XXXX
IN	1011 0XXX XXXX XXXX
SUBI	0101 XXXX XXXX XXXX
SBI	1001 1010 XXXX XXXX

3. Observe the value changes in general purpose registers (GPR), status register (SREG), and program counter (PC) “after” the execution of every instruction. Report your observations in a table.

	Instruction	PC	SREG	R16	R17	R18	R19	R20	R21
1	LDI R16, 0x25	0x00000001	00000000	0x25	0x00	0x00	0x00	0x00	0x00
2	LDI R17, 0x34	0x00000002	00000000	0x25	0x34	0x00	0x00	0x00	0x00
3	LDI R18, 0x31	0x00000003	00000000	0x25	0x34	0x31	0x00	0x00	0x00
4	ADD R16, R17	0x00000004	00000000	0x59	0x34	0x31	0x00	0x00	0x00
5	ADD R16, R18	0x00000005	00001100	0x8A	0x34	0x31	0x00	0x00	0x00
6	LDI R17, 0x74	0x00000006	00001100	0x8A	0x74	0x31	0x00	0x00	0x00
7	ADD R16, R17	0x00000007	00011100	0xFE	0x74	0x31	0x00	0x00	0x00
8	LDI R20, 0	0x00000008	00010100	0xFE	0x74	0x31	0x00	0x00	0x00
9	ADD R16, R20	0x00000009	00010100	0xFE	0x74	0x31	0x00	0x00	0x00
10	LDI R21, 0xFF	0x0000000A	00010100	0xFE	0x74	0x31	0x00	0x00	0xFF
11	OUT DDRD, R21	0x0000000B	00010100	0xFE	0x74	0x31	0x00	0x00	0xFF
12	INC R20	0x0000000C	00000000	0xFE	0x74	0x31	0x00	0x01	0xFF
13	OUT PORTD, R20	0x0000000D	00000000	0xFE	0x74	0x31	0x00	0x01	0xFF
14	STS SUM, R20	0x0000000F	00000000	0xFE	0x74	0x31	0x00	0x01	0xFF
15	RJMP L2	0x0000000B	00000000	0xFE	0x74	0x31	0x00	0x01	0xFF

4. Find out how the value of memory address at 0x300 varies? What do you do if you want to store the value of R20 to the addresses of 0x150 instead of 0x300?

(1) 在0x300的位置，當執行到PC為 0x0000000C到0x0000000F之間時，每一輪執行程式，R20都會在PC為 0x0000000C的步驟加一，並在PC為0x0000000F的步驟，將R20的數值傳送到0x300的位置，也就是每一輪0x300內儲存的數值都加一。

(2) 只要將程式中第一行的程式碼改為.EQU SUM = 0x150，就可以將R20的數值改存在位置0x150中了。

