

✓ **AVR STRUCTURE**

1. Find the organization and chip capacity of each RAM with indicated number of address and data pins.

- a. 11 address , 1 data SRAM → **Org. : $2^{11} * 1$, Capacity : 2 Kbit**
- b. 17 address , 8 data SRAM → **Org. : $2^{17} * 8$, Capacity : 1 Mbit**
- c. 9 address , 1 data DRAM → **Org. : $2^{18} * 1$, Capacity : 256 Kbit**

2. show the ASCII code (in hex) for the following strings :

“U.S.A. is a country” LF

“in North America”

LF is Line Feed

→ **552E532E412E206973206120636F756E7472790A**

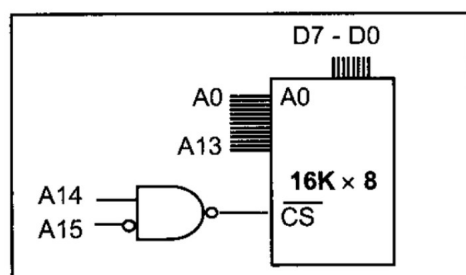
→ **696E204E6F72746820416D6572696361**

.	space	U	S	A	i	s	a	c	o	u	n	t	r	y	LF	N	h	m	e
2E	20	55	53	41	69	73	61	63	6F	75	6E	74	72	79	0A	4E	68	6D	65

3. A given computer has a 32-bit data bus. What is the largest number that can be carried into the CPU at a time?

→ **the largest number is $2^{32}-1$**

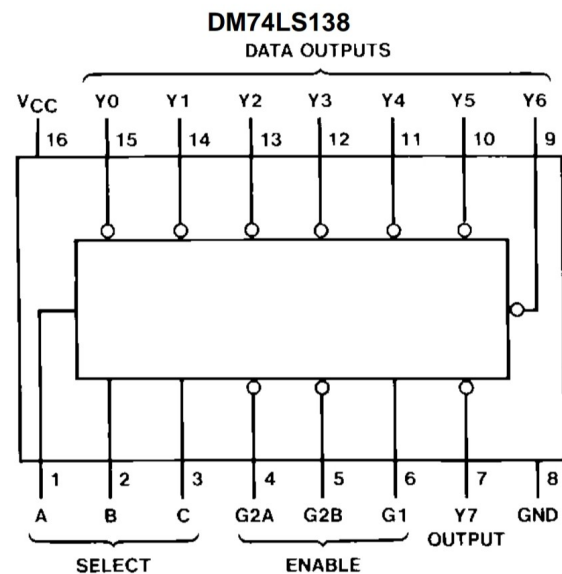
4. Find the address range of memory design in the diagram



→ So the address range could be :
[0x4000 , 0x7FFF]

5. Using the 74138, design the memory decoding circuitry in which the memory block controlled by Y0 is in the range 0000 Hex to 1FFF Hex. Indicate the size of the memory block controlled by each Y.

MEMORY BLOCK CONTROLLER	SELECT ADDRESS BUS				A11 ... A0
	A15	A14	A13	A12	
	Y0	0	0	0	0x000 0xFFFF
	Y1	0	0	1	0x000 0xFFFF
	Y2	0	1	0	0x000 0xFFFF
	Y3	0	1	1	0x000 0xFFFF
	Y4	1	0	0	0x000 0xFFFF
	Y5	1	0	1	0x000 0xFFFF
	Y6	1	1	0	0x000 0xFFFF
	Y7	1	1	1	0x000 0xFFFF



با توجه به جدول فوق هر Y میتواند 8Kbit را کنترل کند پس در مجموع یک حافظه 64Kbit داریم.

✓ INSTRUCTION SET

6. Which of the following is (are) illegal, and why?

- a) ADD R20, R11 صحیح است
- b) ADD R16, R1 صحیح است
- c) ADD R52, R16 غلط است زیرا R52 نداریم
- d) LDI R16, \$255 غلط است زیرا \$255 را نمیتوان در یک بایت گنجاند
- e) LDI R23, 0xF5 صحیح است

در انتها فرم درست دو دستور ADD و LDI را بیان میکنم :

ADD Rd , Rr \equiv Rd \leftarrow Rd + Rr ; $0 \leq d \leq 31$, $0 \leq r \leq 31$

0000	11rd	dddd	rrrr
------	------	------	------

LDI Rd , K \equiv Rd \leftarrow K ; $16 \leq d \leq 31$, $0 \leq K \leq 255$

1110	kkkk	dddd	kkkk
------	------	------	------

7. What is the status of the C and Z flags after the following code?

LDI R20, 0xFF \rightarrow **C = 0 , Z = 0**

LDI R21, 1 \rightarrow **C = 0 , Z = 0**

ADD R20, R21 \rightarrow **C = 1 , Z = 1**

8. Find the C flag value after each of the following codes:

(a) LDI R20, 0x54 \rightarrow **C = 0**

LDI R25, 0xC4 \rightarrow **C = 0**

ADD R20, R25 \rightarrow **C = 1**

(b) LDI R23, 0 \rightarrow **C = 0**

LDI R16, 0xFF \rightarrow **C = 0**

ADD R23, R16 \rightarrow **C = 0**

(c) LDI R30, 0xFF \rightarrow **C = 0**

LDI R18, 0x05 \rightarrow **C = 0**

ADD R30, R18 \rightarrow **C = 1**

9. Indicate the size (8 or 16 bit) of each of the following registers.

R0= **8 bit**

R24= **8 bit**

Data memory Location \$300= **8 bit**

PORTA= **8 bit**

Program memory Location \$300= **16 bit**

10. Indicate the largest value (in decimal) that each register can contain.

R0= **(signed = 127) and (unsigned = 255)**

R24= **(signed = 127) and (unsigned = 255)**

Data memory Location \$300= (signed = 127) and (unsigned = 255)
 PORTA= (signed = 127) and (unsigned = 255)
 Program memory Location \$300= $2^{16} - 1 = 65535$

11. Who generates each of the following files and what is the use of each.

a. .asm b. .lst c. .obj d. .eep e. .hex

تمام فایل های فوق به جز .asm که توسط خود برنامه نویسی ایجاد میشود ، بقیه فایل ها توسط اسمبلر ایجاد میشوند.

1. فایل .asm که حاوی کدهای اسمبلی برنامه نویسی است (برنامه ای که باید توسط میکرو اجرا شود) و تغذیه ی اسمبلر برای تهیه سایر فایل هاست.

2. فایل .lst شامل اطلاعات بسیار زیادی درباره پروسه ی اسمبلی میشود و بخش هایی دارد مثل warnings and errors , page header , command line , source listing , symbol table , cross reference

3. فایل .obj یک فایل با فرمت خاص که شامل رکوردهای دیتایی درباره symbolic information , object code , debugging information , library reference and so on دارد ، همچنین این فایل یک مرجع به شماره خط داخل برنامه اصلی ایجاد میکند ولی شامل خود کد اصلی نیست

4. فایل .eep شامل اطلاعات EEPROM میباشد

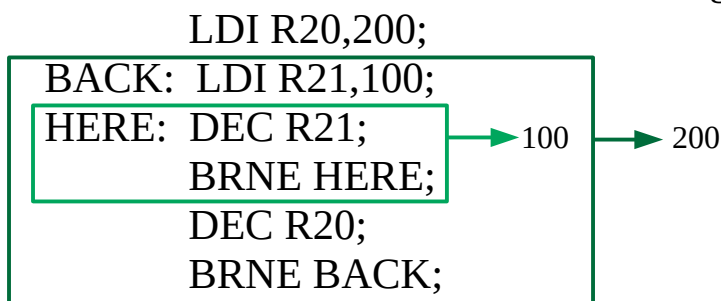
5. فایل .hex شامل اطلاعات FLASH میباشد.

به سلسله مراتب زیر توجه کنید :

source lang. → (compiler) → .o → (linker) → .elf → (obj-drump disassembler) → .lss

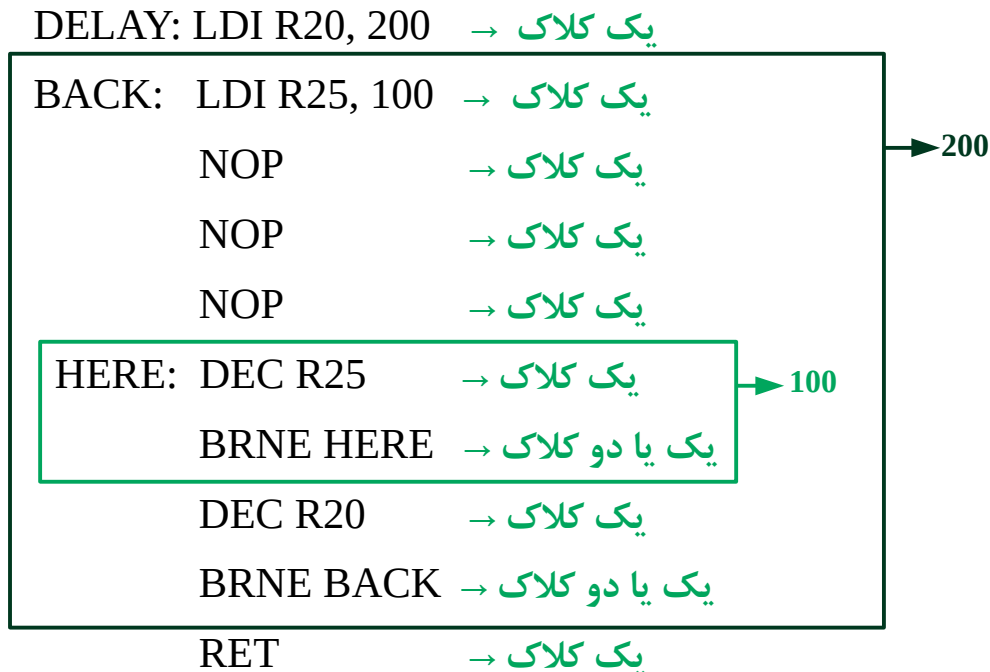
✓ **PROGRAMMING I**

12. Find the number of times the following loop is performed:



← در مجموع این حلقه ی تودرتو 100*200 یعنی 20000 بار تکرار میشود

13. Find the time delay for the delay subroutine shown below if the system has an AVR with a clock frequency of 20 MHz.



در مجموع این subroutine به اندازه ی:

$$1 + 200*(1+1+1+1+1+2) - 1 + 200*(100*(1+2) - 1) + 1 = 161201$$

زمان واقعی که این subroutine تاخیر ایجاد میکند برابر است با :

$$161201 * 1/(20\text{MHz}) = 8060.05 \mu\text{sec} \approx 8 \text{ msec}$$

14. What is the value of R20 after the code below is finished?

LDI R20, 0x40 ; **R20 ← 0x40**
 CLC ; **clear carry** $\equiv C \leftarrow 0$
 ROR R20 ; **R20₇ ← C , R20_n ← R20_{n+1} , C ← R20₀ \equiv R20 ← 0x20**
 ROR R20 ; **R20 ← 0x10**
 ROR R20 ; **R20 ← 0x08**
 ROR R20 ; **R20 ← 0x04**
 SWAP R20 ; **R20_{3...0} ↔ R20_{7...4} \equiv R20 ← 0x40**

در نهایت ، مقدار رجیستر R20 برابر با 0x40 خواهد شد، یعنی عملاً این کد هیچ تغییری در مقدار اولیه این رجیستر ایجاد نکرد !

15. What is the value of R20 after the code below is finished?

```
LDI R20 , 0x00 ; R20 ← 0x00
SEC ; set carry ≡ C ← 1
ROL R20 ; R200 ← C , R20n+1 ← R20n , C ← R207 ≡ R20 ← 0x01
CLC ; clear carry ≡ C ← 0
ROL R20 ; R20 ← 0x02
SEC ; set carry ≡ C ← 1
ROL R20 ; R20 ← 0x05
CLC ; clear carry ≡ C ← 0
ROL R20 ; R20 ← 0x0A
SEC ; set carry ≡ C ← 1
ROL R20 ; R20 ← 0x15
CLC ; clear carry ≡ C ← 0
ROL R20 ; R20 ← 0x2A
SEC ; set carry ≡ C ← 1
ROL R20 ; R20 ← 0x55
CLC ; clear carry ≡ C ← 0
ROL R20 ; R20 ← 0xAA
```

در نهایت ، مقدار رجیستر R20 برابر با 0xAA خواهد شد.

✓ **PROGRAMMING II**

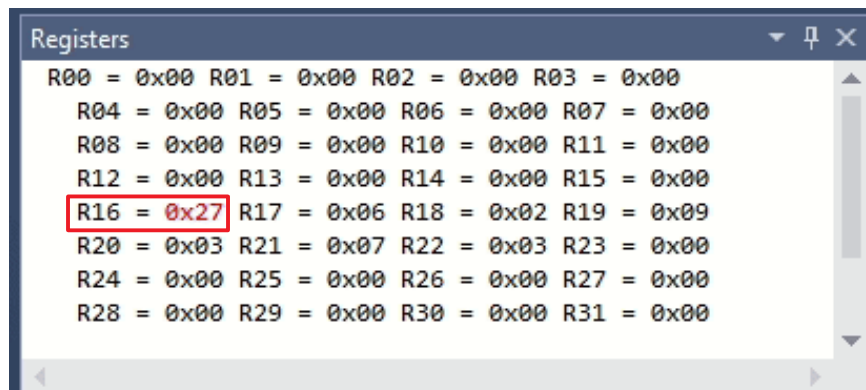
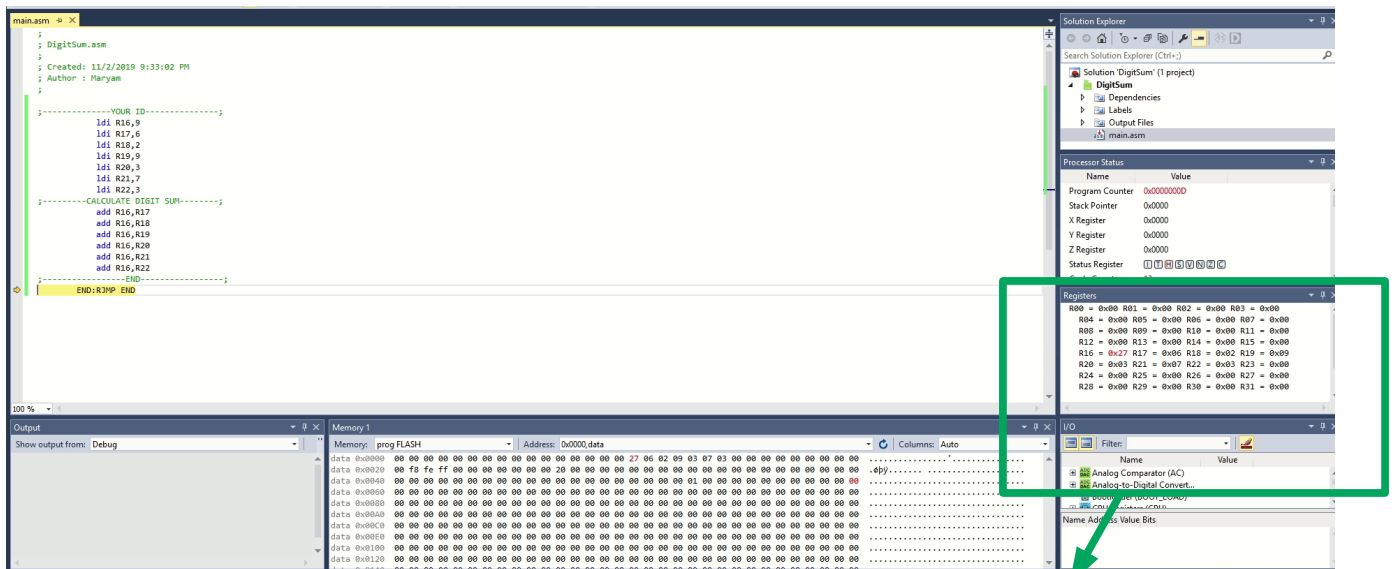
16. Write and assemble a program to add all the single digits of your ID number and save the result in R16. Simulate this program in AVR studio and check it.

```
;
; DigitSum.asm
;
; Created: 11/2/2019 9:33:02 PM
; Author : Maryam
```

```

;
;-----YOUR ID-----;
    ldi R16,9
    ldi R17,6
    ldi R18,2
    ldi R19,9
    ldi R20,3
    ldi R21,7
    ldi R22,3
;-----CALCULATE DIGIT SUM-----;
    add R16,R17
    add R16,R18
    add R16,R19
    add R16,R20
    add R16,R21
    add R16,R22
;-----END-----;
END:RJMP END

```



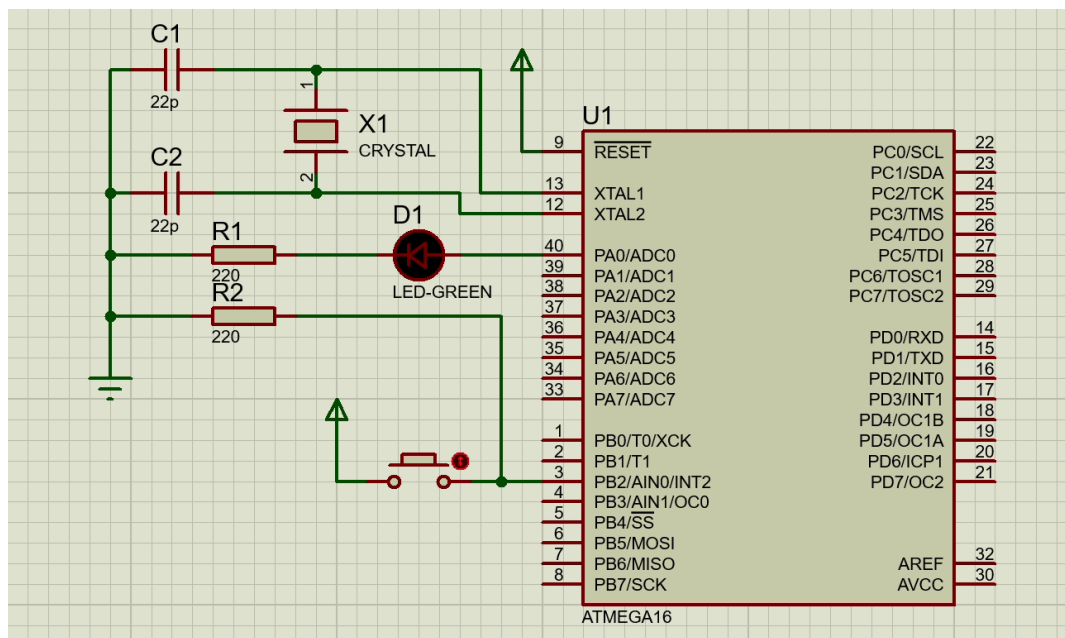
همانطور که در تصویر پیداست ، مقدار نهایی رجیستر 16 برابر با 0x27 که معادل 39 است زیرا :

$$9 + 6 + 2 + 9 + 3 + 7 + 3 = 39$$

17. Suppose a switch is connected to the Pin 2 of PORTB. Write an algorithm to get its state, using the debouncing algorithm. Simulate this program in AVR studio, draw the circuit in Proteus and check it.

* الگوریتم debouncing :

1. لحظه اولی که فشرده شدن کلید تشخیص داده شده
2. صبر به اندازه ی 20msec
3. دوباره کلید چک می شود ، اگر وضعیت آن مشابه (1) بود ، یعنی کلید واقعا فشار داده شده است و به (4) برو در غیر این صورت به (1) برو.
4. صبر می کنیم تا کلید رها شود.
5. صبر به اندازه ی 20msec
6. چک می کنیم که آیا واقعا کلید رها شده یا نه .



← لطفا برای توضیحات بیشتر به این [لینک](#) مراجعه کنید.

✓ **SIMULATION & IMPELEMENTATION (GROUP QUESTION)**

18. Design and make a circuit with ATmega32, in which there is 8 LEDs. LEDs turn on, one by one. There is a switch, by pressing it, the blinking process starts from right to left, and continue. By pressing the switch once again, direction of LED turning on changes.

← لطفا برای توضیحات بیشتر به این [لینک](#) مراجعه کنید.