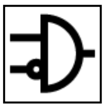
2018-10-29, Microprocessor Midterm Exam

* Please solve the following problems in the answer sheet. (답안은 답안지에 작성해 주시기 바랍니다.)

* You must show your work unless the answer is obvious. (풀이 과정이 없는 답은 인정하지 않습니다.)

Student ID	Name	Signature

--------------------Lecture Problems--



1. (10) (a) (5) Explain the three kinds of buses on 8051 with their directions.

   (b) (5) Explain the meaning of Universal gates, and prove that (A & ~B) gates are universal gates.

2. (10) (a) (5) What are the two core characteristics of "new memory(NVM-based)" in computer systems? (Hint: please compare to DRAM and Storage)

   (b) (5) When two different 8051-compatible processors with same external clock frequency are used, can we say that "performance of the two processors are same?" If not, please explain possible reasons. 2021

3. (10pt) (a) (5) Which of the following is (are) illegal?

   (1) MOV R1, #600 (2) MOV R2, #30 (3) MOV A, #100H (4) MOV A, #F5H (5) MOV R12, #50H

   (b) (5) How much does PC (program counter) increase when executing an instruction? (Is this a fixed amount or not?)

4. (10) (a) (5) Which memory space will be used if DB (defined byte) directive is used for a constant value? Please explain your answer. (Hint: ROM/RAM/None of them)

   (b) (5) If a programmer decides to use register bank 1, which additional action should be added to use stack operations correctly? Please explain your answer.

5. (10) Show the contents of the register R0~R7 and SP after finishing the program execution. (default setting)

1 (instruction #)	ORG 0	5	MOV R7, #01H	9	POP 3
2	MOV R0, #66H	6	PUSH 2	10	POP 4
3	MOV R1, #7FH	7	PUSH 1	11	POP 5
4	MOV R2, #5DH	8	PUSH 0	12	POP 6

12

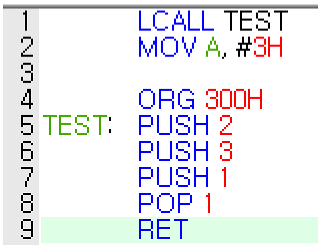
6. (10) (a) (5) Find the values of R1 and A registers after execution of the following program.

   (b) (5) Rewrite this program using only one DJNZ instruction.

```
LOC   OBJ            LINE      SOURCE
0000 7900              1       MOV R1, #0H
0002 7455              2       MOV A, #55H
0004 7B05              3       MOV R3, #5H
0006 7A05              4       LOOP2: MOV R2, #5H
0008 F4                5       LOOP1: CPL A
0009 09                6       INC R1
000A DA( 7.(a) )       7       DJNZ R2, LOOP1
000C DB( 7.(a) )       8       DJNZ R3, LOOP2
```

7. (10) (a) (5) For the program of Problem 6, what are the OBJs for DJNZ R2, LOOP1 and DJNZ R3, LOOP2? (Hint: DA(), and DB())

   (b) (5) Assume that the values of all general registers (Rn) are Zero(0), and the following program starts from ORG 0. This program does not work correctly. Why?



```
        LCALL TEST
        MOV A, #3H

        ORG 300H
TEST:   PUSH 2
        PUSH 3
        PUSH 1
        POP 1
        RET
```

8. (10) (a) (5) When we execute three LCALL instructions without any RET instruction, what is the current value of SP register with default setting?

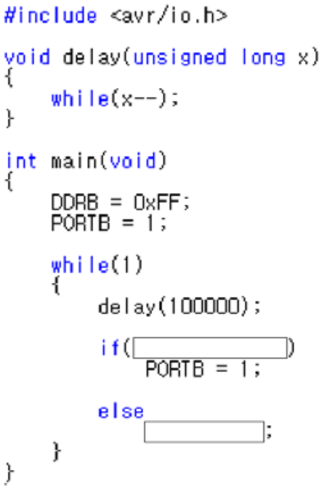
   (b) (5) If we don't know the I/O mode of P1, and would like to use P1 as Input mode, what should we do?

9. (10) When assuming that we have a DELAY subroutine, write a program to create a square wave of 75% duty cycle on bit 3 of port 3 using the DELAY subroutine. (You don't need to write DELAY subroutine)

---------------------Practice Problems--

10. (10) (a) (5) In AVR Atemega128, explain about Data Direction register, PORT register, and PIN register, including explanation of the functionality when it is set to 0 and 1.

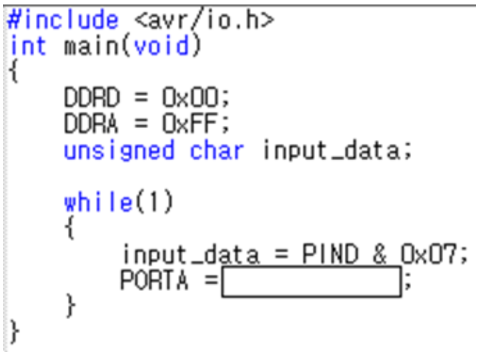
    (b) (5) Fill in the blanks of the following program that turns on one LED from 0 to 7 sequentially on an 8-bit LED module. (Only one LED should be turned on at a time and LED 0 is turned on again after LED 7.)



```
#include <avr/io.h>

void delay(unsigned long x)
{
    while(x--);
}

int main(void)
{
    DDRB = 0xFF;
    PORTB = 1;

    while(1)
    {
        delay(100000);

        if(□)
            PORTB = 1;

        else □;
    }
}
```

11. (10) (a) (5) Explain how to display "1234" in a Dynamic FND (an array of 4 Static FNDs (7-segments)) in our practice board environment.

    (b) (5) Fill in the blanks when MCU PIND receives lower 3 bits from 8-to-3 encoder and decodes it to PORTA.



```
#include <avr/io.h>
int main(void)
{
    DDRD = 0x00;
    DDRA = 0xFF;
    unsigned char input_data;

    while(1)
    {
        input_data = PIND & 0x07;
        PORTA =□;
    }
}
```

Arithmetic operations	
ADD	A,Rn
ADD	A,direct
ADD	A,@RI
ADD	A,#data
ADDC	A,Rn
ADDC	A,direct
ADDC	A,@RI
ADDC	A,#data
SUBB	A,Rn
SUBB	A,direct
SUBB	A,@RI
SUBB	A,#data
INC	A
INC	Rn
INC	direct
INC	@RI
DEC	A
DEC	Rn
DEC	direct
DEC	@RI
INC	DPTR
MUL	AB
DIV	AB
DA	A
CLR	A
CPL	A
RL	A
RLC	A
RR	A
RRC	A
SWAP	A

Program and machine c	
ACALL	addr11
LCALL	addr16
RET	
RETI	
AJMP	addr11
LJMP	addr16
SJMP	rel
JMP	@A+DPTR
JZ	rel
JNZ	rel
JC	rel
JNC	rel
JB	bit,rel
JNB	bit,rel
JBC	bit,rel
CJNE	A,direct,rel

jne A,#data,@
(jump if A ! = data)

je A, #data,@
(jump if A == data)

ja, jnbe A,#data,@
(jump if A > data)

jae, jnb A,#data,@
(jump if A >= data)

jb, jnae A,#data,@
(jump if A < data)

jbe, jna A,#data,@
(jump if A <= data)

switch A <,==,> #data
(no A modification)

Data transfer	
MOV	A,Rn
MOV	A,direct*)
MOV	A,@RI
MOV	A,#data
MOV	Rn,A
MOV	Rn,direct
MOV	Rn,#data
MOV	direct,A
MOV	direct,Rn
MOV	direct,direct
MOV	direct,@RI
MOV	direct,#data
MOV	@RI,A
MOV	@RI,direct
MOV	@RI,#data
MOV	DPTR,#data16
MOVC	A,@A+DPTR
MOVC	A,@A+PC
MOVX	A,@RI
MOVX	A,@DPTR
MOVX	@RI,A
MOVX	@DPTR,A
PUSH	direct
POP	direct
XCH	A,Rn
XCH	A,direct
XCH	A,@RI
XCHD	A,@RI
CJNE	A,#data,rel
CJNE	Rn,#data,rel
CJNE	@Rn,#data,rel
DJNZ	Rn,rel
DJNZ	direct,rel
NOP	

Logic operations	
ANL	A,Rn
ANL	A,direct
ANL	A,@RI
ANL	A,#data
ANL	direct,A
ANL	direct,#data
ORL	A,Rn
ORL	A,direct
ORL	A,@RI
ORL	A,#data
ORL	direct,A
ORL	direct,#data
XRL	A,Rn
XRL	A,direct
XRL	A,@RI
XRL	A,#data
XRL	direct,A
XRL	direct,#data

Boolean variable manip	
CLR	C
CLR	bit
SETB	C
SETB	bit
CPL	C
CPL	bit
ANL	C,bit
ANL	C,/bit
ORL	C,bit
ORL	C,/bit
MOV	C,bit
MOV	bit,C