

학과		학번		이름	
----	--	----	--	----	--

1. Mano의 Basic Computer에서의 인터럽트 서비스를 스택을 이용하는 방식으로 할 때, 다음 물음에 답하라. 스택에 저장될 위치를 알려주는 값은 스택포인터가 가지고 있다. 스택은 메모리의 4001번지부터 시작되고, 스택 포인터는 메모리 4000번지에 저장된다. [총 20점]

LINE	PROGRAM			LINE	PROGRAM		
1		ORG	0	27		BUN	EPI I
2	TOP,	HEX	0	28	DSP,	HEX	0
3		SKO		29		LDA	SP
4		BUN	①	30		ADD	④
5		BUN	OUT	31		STA	SP
6	PRO,	HEX	0	32		BUN	DSP I
7		STA	SP I	33	MIN,	DEC	-1
8		ISZ	SP	34	INP,	IOF	
9		CIR		35		BSA	⑤
10		STA	SP I	36		INP	
11		ISZ	SP	37		OUT	
12		LDA	TOP	38		STA	CHI
13		STA	SP I	39		BSA	EPI
14		ISZ	SP	40		BUN	TOP I
15		②		41	OUT,	IOF	
16		BUN	PRO I	42		BSA	PRO
17	EPI,	HEX	0	43		ION	
18		BSA	DSP	44		LDA	CHO
19		LDA	SP I	45		OUT	
20		STA	TOP	46		IOF	
21		BSA	DSP	47		BSA	EPI
22		LDA	SP I	48		BUN	TOP I
23		③		49	CHI,	HEX	0
24		BSA	DSP	50	CHO,	HEX	0
25		LDA	SP I	51		ORG	4000
26		ION		52	SP,	DEC	4001

(1) ① ~ ⑤ 에 들어갈 명령어나 값, 혹은 변수명을 써라. [각 2점]

① INP	② ION	③ CIL	④ MIN	⑤ PRO
-------	-------	-------	-------	-------

(2) 사용자 프로그램과 위 루틴이 공유할 수 있는 값은 레지스터들의 값과 스택의 시작주소이다. 공유되는 값을 가지고 있는 레지스터(또는 flip-flop)들의 이름을 적으시오. [각 2점]

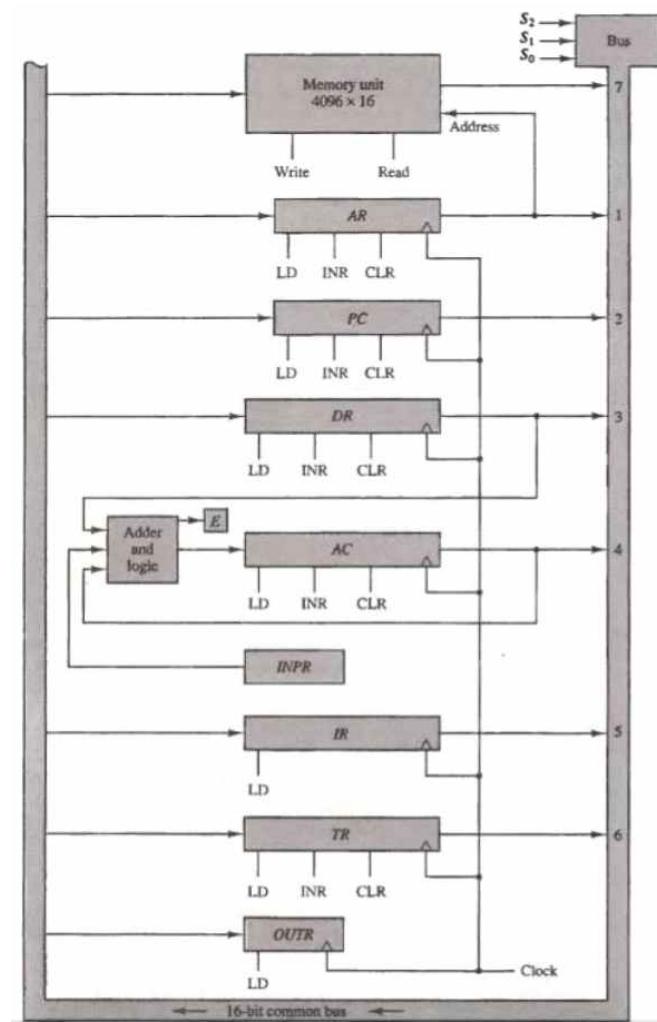
AC	E
----	---

(3) 서브루틴 DSP 대신에 DSZ라는 memory-reference instruction으로 대신하려고 한다. DSZ는 ISZ와 반대로 주어진 메모리 주소에 있는 값을 1 감소시킨 다음 그 메모리 주소로 다시 저장하

고 감소시킨 값이 0이면 다음 instruction을 skip한다. DSZ명령어를 위한 RTL을 작성하라. (단, 명령어 실행 후 원래 AC에 저장되어 있던 값이 변경되어서는 안되며, DSZ명령어는 15 clock 이내에 수행되어야 한다) [6점]

D <sub>6</sub> T <sub>4</sub> : TR ← AC	D <sub>6</sub> T <sub>4</sub> : DR ← M[AR]
D <sub>6</sub> T <sub>5</sub> : DR ← M[AR], AC ← 0	D <sub>6</sub> T <sub>5</sub> : AC ← DR, DR ← AC
D <sub>6</sub> T <sub>6</sub> : AC ← complement AC	D <sub>6</sub> T <sub>6</sub> : AC ← complement AC
D <sub>6</sub> T <sub>7</sub> : AC ← AC + DR	D <sub>6</sub> T <sub>7</sub> : AC ← AC + 1
D <sub>6</sub> T <sub>8</sub> : M[AR] ← AC, if (AC = 0) then (PC ← PC + 1)	D <sub>6</sub> T <sub>8</sub> : AC ← complement AC
D <sub>6</sub> T <sub>9</sub> : DR ← TR	D <sub>6</sub> T <sub>9</sub> : M[AR] ← AC, if (AC = 0) then (PC ← PC + 1)
D <sub>6</sub> T <sub>10</sub> : AC ← DR, SC ← 0	D <sub>6</sub> T <sub>10</sub> : AC ← DR, SC ← 0

2. 아래의 Mano가 설계한 컴퓨터의 버스구조와 RTL을 보고 다음 물음에 답하라. [총 20점]



(1) 메모리 위치 X, Y에 저장된 두 수를 OR하는 프로그램 모듈을 작성하라. [3점]

LINE	프로그램	LINE	프로그램
1	X, HEX 10	7	LDA Y
2	Y, HEX 20	8	CMA
3	TMP, HEX 0	9	AND TMP
4	LDA X	10	CMA
5	CMA	11	
6	STA TMP	12	

- (2) 위 문제를 통하여 제시된 컴퓨터의 명령어에 더 이상의 추가 명령어가 없어도 모든 프로그램을 작성할 수 있음을 나타내는 용어는 무엇인가? [2점]

**Instruction set completeness**

- (3) 레지스터 AR의 제어선(LD, INR, CLR)에 연결될 회로를 위한 Boolean 수식을 RTL의 제어필드를 이용하여 표현하라. [각 2점]

LD	$R' T_0 + R' T_2 + D_7' IT_3$
INR	$D_5 T_4$
CLR	$RT_0$

- (4) BUS의 제어선( $S_0 \sim S_2$ )에 연결될 회로를 위한 Boolean 수식을 RTL의 제어필드를 이용하여 표현하라. [각 2점]

$S_0$	$R' T_1 + R' T_2 + D_7' IT_3 + (D_0 + D_1 + D_2 + D_4 + D_6) T_4 + D_5 T_5 + D_6 T_6$
$S_1$	$T_0 + T_1 + D_7' IT_3 + (D_0 + D_1 + D_2 + D_5 + D_6) T_4 + D_6 T_6$
$S_2$	$T_1 + R' T_2 + D_7' IT_3 + (D_0 + D_1 + D_2 + D_3 + D_6) T_4 + pB_{10}$

- (5) LDA 명령어의  $D_2 T_4$ 와  $D_2 T_5$ 를 기준의 다른 명령어들에 사용된 RTL을 활용하여 다시 작성 하라. [3점]

$D_2 T_4 : DR \leftarrow M[AR], AC \leftarrow 0$   
 $D_2 T_5 : AC \leftarrow AC + DR, SC \leftarrow 0$

3. 아래 글을 읽고 물음에 답하라. [총 20점]

The process of communication is done by programmed control transfer which is implemented by ①the loop of instructions in Mano's architecture. ②The computer keeps checking the flag bits for input and output alternately, and when it finds one of them set, it initiates an information transfer. The difference of information flow rate between the computer and that of the input-output devices makes this type of transfer inefficient. To see why this is inefficient, consider a computer that can executes ③one MIPS. Usually man can type keyboard at a maximum rate of 10 ④CPS. The printer can transfer information at a maximum rate of 20 ⑤LPM with 30 characters per line. This means that at the maximum rate, the computer will check the flags ⑥□ times between each transfer. The computer is ⑦wasting time while checking the flag instead of doing some other useful processing task.

(1) ①에 해당하는 입력을 위한 프로그램 모듈을 작성하라. [2점]

CIF, SKI  
BUN CIF

(2) ②에 해당하는 용어를 무엇이라고 하는가? [2점]

Polling

(3) ③을 근거로 computer의 instruction cycle time을 계산하라. [2점]

$10^{-6}$  second

(4) ④, ⑤의 약자가 아닌 전체 용어를 작성하라. [각 2점]

CPS : Characters Per Second

LPM : Lines Per Minute

(5) 한 개의 문자를 입력하는 동안 몇 번의 instruction cycle이 수행되는가? [1점]

$10^5$

(6) 프린터가 한 글자를 출력하는 동안 몇 번의 instruction cycle이 수행되는가? [1점]

$10^5$

(7) ⑦처럼 하나의 글자를 입출력하기 위하여 낭비되는 instruction cycle은 얼마인가? [2점]

$2 \times 10^5$

(8) 이런 낭비를 줄이는 방법은 무엇인가? [2점]

Interrupt

(9) 위 (8)의 방법에서 입출력을 전달하는 프로그램을 무엇이라 부르는가? [2점]

I/O Service Program

(10) 위 방법을 구현하는 2번의 Mano's Architecture에서 IEN flag의 기능은 무엇인가? [2점]

Interrupt facility의 사용 유무를 결정

4. 다음 프로그램은 16비트 두 정수의 곱셈을 수행하는 프로그램이다. [총 20점]

LINE	PROGRAM			LINE	PROGRAM		
1		ORG	100	36		INC	
2		CLE		37		STA	SGN
3		LDA	FST	38		BUN	SUB
4		STA	XP	39	PN,	LDA	YN
5		CMA		40		STA	SND
6		INC		41		LDA	SGN
7		STA	XN	42		INC	
8		LDA	SND	43		STA	SGN
9		STA	YP	44		BUN	SUB
10		CMA		45	PP,	BUN	SUB
11		INC		46	// MULTIPLICATION		
12		STA	YN	47	SUB,	LDA	N
13	// SIGN CHECK			48		BSA	MUL
14		LDA	FST	49	FST,	HEX	0
15		CIL		50	SND,	HEX	0
16		SZE		51	TRD,	HEX	0
17		BUN	NEG	52	FOR,	HEX	0
18		LDA	SND	53		LDA	SGN
19		CIL		54		CIR	
20		SZE		55		SZE	
21		BUN	PN	56		BUN	POS
22		BUN	PP	57		CLE	
23	NEG,	LDA	SND	58		LDA	TRD
24		CIL		59		CMA	
25		SZE		60		STA	TRD
26		BUN	NN	61		LDA	FOR
27		BUN	NP	62		CMA	
28	NN,	LDA	XN	63		STA	FOR
29		STA	FST	64		CLA	
30		LDA	YN	65		INC	
31		STA	SND	66		ADD	FOR
32		BUN	SUB	67		STA	FOR
33	NP,	LDA	XN	68		CLA	
34		STA	FST	69		CIL	
35		LDA	SGN	70		ADD	TRD

LINE	PROGRAM			LINE	PROGRAM		
71		STA	TRD	100		CIL	
72	POS,	HLT		101		ADD	XH
73	N,	DEC	-16	102		ADD	PH
74	SGN,	DEC	-1	103		STA	PH
75	XP,	HEX	0	104		CLE	
76	XN,	HEX	0	105	ZRO,	LDA	XL
77	YP,	HEX	0	106		CIL	
78	YN,	HEX	0	107		STA	XL
79		ORG	200	108		LDA	XH
80	MUL,	HEX	0	109		CIL	
81		STA	CTR	110		STA	XH
82		LDA	MUL I	111		ISZ	CTR
83		STA	XL	112		BUN	LOP
84		ISZ	MUL	113		LDA	PH
85		LDA	MUL I	114		STA	MUL I
86		STA	YL	115		ISZ	MUL
87		ISZ	MUL	116		LDA	PL
88	LOP,	CLE		117		STA	MUL I
89		LDA	YL	118		ISZ	MUL
90		CIR		119		BUN	MUL I
91		STA	YL	120	CTR,	DEC	0
92		SZE		121	XH,	HEX	0
93		BUN	ONE	122	XL,	HEX	0
94		BUN	ZRO	123	YH,	HEX	0
95	ONE,	CLE		124	YL,	HEX	0
96		LDA	XL	125	PH,	HEX	0
97		ADD	PL	126	PL,	HEX	0
98		STA	PL	127		END	
99		CLA		128			

(1) 실제 곱셈이 될 두 수가 저장된 위치에 해당하는 변수를 적으시오. [각 2점]

FST	SND
-----	-----

(2) (1)의 두 값이 21과 -16이라고 할 때, 프로그램이 실행되는 과정에서 XL과 YL이 어떻게 변하는지 세 단계까지 적으시오. (값은 16진법으로 표기한다) [각 0.5점]

XL	0	->	15	->	2A	->	54
YL	0	->	10	->	8	->	4

(3) 위 (2)번 문제를 제시된 프로그램 대신에 Booth 알고리즘을 사용하는 경우 효율적인 이유를 실제 숫자를 이용하여 설명하라. [2점]

Booth 알고리즘을 사용하는 경우 두 수의 부호와 상관없이 곱셈이 가능하며, 특히 $-16 = -2^4$ 으로 표현되기 때문에 곱셈 과정에서 덧셈 연산을 한 번만 사용하여 효율적이다.
---

- (4) (1)의 두 값이 21과 -16이라고 할 때, 프로그램이 실행된 이후 최종 결과가 저장되는 변수명들과 그 결과 값을 16진법으로 표시하라. [1점]

변수명	TRD	FOR
결과값	FFFF	FEB0

- (5) 22번 문장과 45번 문장은 하나의 문장으로 바꾸어도 프로그램의 수행에 지장이 없다. 수정된 하나의 문장의 번호와 명령어를 작성하라. [각 1점]

LINE	PROGRAM		
22		BUN	SUB

- (6) 위 프로그램에서 두 수 중 하나라도 값이 0(zero)이면 결과 값은 0이 되므로 이 경우에 프로그램을 바로 끝내도록 하면 곱셈을 보다 효율적으로 수행할 수 있다. 이를 위하여 3번과 4번 사이, 8번과 9번 사이에 다음과 같은 명령어들을 추가한다고 하였을 때, 아래의 프로그램 모듈을 완성하라. [각 1점]

3.1		SZA	
3.2		BUN	NXT
3.3		① BUN	② POS
4	NXT,	STA	XP
8.1		③ SZA	
8.2		BUN	NEX
8.3		① BUN	② POS
9	NEX,	STA	YP

5. 다음 프로그램은 부동소수점의 실수원소를 가지는 배열 A와 배열 B를 더하여서 배열 C에 저장하는 프로그램이다. 배열에는 sign 1비트, exponent 3비트, mantissa 12비트가 한 word 씩 차례대로 저장되며, 최초에 mantissa는 normalized된 상태로 저장된다. [총 20점]

LINE	PROGRAM			LINE	PROGRAM		
1		ORG	200	36		STA	E1
2	LOP,	LDA	PT1 I	37		BUN	CP1
3		STA	S1	38	L3,	LDA	M2
4		ISZ	PT1	39		CLE	
5		LDA	PT1 I	40		CIR	
6		STA	E1	41		STA	M2
7		ISZ	PT1	42		LDA	E2
8		LDA	PT1 I	43		INC	
9		STA	M1	44		STA	E2
10		ISZ	PT1	45		BUN	CP1
11		LDA	PT2 I	46	CP2,	LDA	S2
12		STA	S2	47		CMA	
13		ISZ	PT2	48		INC	
14		LDA	PT2 I	49		ADD	S1
15		STA	E2	50		SZA	
16		ISZ	PT2	51		BUN	L4
17		LDA	PT2 I	52		BUN	L5
18		STA	M2	53	L4,	CLE	
19		ISZ	PT2	54		LDA	M2
20	CP1,	LDA	E2	55		CMA	
21		CMA		56		INC	
22		INC		57		ADD	M1
23		ADD	E1	58		STA	M1
24		SZA		59		SZE	
25		BUN	L1	60		BUN	L6
26		BUN	CP2	61		LDA	M1
27	L1,	SPA		62		CMA	
28		BUN	L2	63		INC	
29		BUN	L3	64		STA	M1
30	L2,	LDA	M1	65		LDA	S1
31		CLE		66		CMA	
32		CIR		67		AND	MSK
33		STA	M1	68		STA	S1
34		LDA	E1	69	L6,	CLE	
35		INC		70		LDA	M1

LINE	PROGRAM			LINE	PROGRAM		
71		CIL		106	CTR,	DEC	-2
72		SZE		107	S1,	HEX	0
73		BUN	FIN	108	E1,	HEX	0
74		STA	M1	109	M1,	HEX	0
75		LDA	E1	110	S2,	HEX	0
76		ADD	MIN	111	E2,	HEX	0
77		STA	E1	112	M2,	HEX	0
78		BUN	L6	113	MSK,	HEX	0001
79	L5,	CLE		114	MIN,	HEX	FFFF
80		LDA	M2	115		ORG	300
81		ADD	M1	116		HEX	0
82		STA	M1	117		HEX	2
83		SZE		118		HEX	D000
84		BUN	L7	119		HEX	0
85		BUN	FIN	120		HEX	2
86	L7,	CIR		121		HEX	D000
87		STA	M1	122		ORG	400
88		LDA	E1	123		HEX	0
89		INC		124		HEX	4
90		STA	E1	125		HEX	8000
91	FIN,	LDA	S1	126		HEX	1
92		STA	PT3 I	127		HEX	4
93		ISZ	PT3	128		HEX	8000
94		LDA	E1	129		ORG	500
95		STA	PT3 I	130		HEX	0
96		ISZ	PT3	131		HEX	0
97		LDA	M1	132		HEX	0
98		STA	PT3 I	133		HEX	0
99		ISZ	PT3	134		HEX	0
100		ISZ	CTR	135		HEX	0
101		BUN	LOP	136		END	
102		HLT		137			
103	PT1,	HEX	300	138			
104	PT2,	HEX	400	139			
105	PT3,	HEX	500	140			

(1) 각 배열의 시작 주소가 저장된 변수명을 써라. [각 1점]

PT1	PT2	PT3
-----	-----	-----

(2) 프로그램에서 부동소수점 덧셈을 위해 두 배열의 sign, exponent, mantissa 값이 저장되는 변수명을 써라. [각 0.5점]

	sign	exponent	mantissa
배열 A	S1	E1	M1
배열 B	S2	E2	M2

(3) 위 프로그램은 크게 mantissa alignment  $\rightarrow$  mantissa addition(or subtraction)  $\rightarrow$  normalization 순서로 수행된다. 프로그램의 20 ~ 45라인은 위의 세 과정 중 어느 과정에 해당하는지 적으시오. [3점]

mantissa alignment

(4) 두 배열 A, B를 더하는 횟수를 결정하는 변수명과 해당 변수의 초기 값을 적으시오. [각 1점]

변수 명 : CTR	초기 값 : -2
------------	-----------

(5) 두 부동소수점 수에 대한 덧셈 혹은 뺄셈의 수행 결과로 overflow 또는 underflow가 발생할 수 있다. 위 프로그램에서는 각각의 경우에 대해 어떠한 처리를 수행하는가? [각 2점]

overflow일 때는 mantissa를 shift right한 뒤 exponent 값을 1 증가,  
underflow일 때는 mantissa를 shift left한 뒤 exponent 값을 1 감소

(6) 변수 MSK의 역할을 설명하라. [2점]

sign값을 0에서 1로, 또는 1에서 0으로 바꾸기 위해 사용

(7) 프로그램의 수행이 끝나고 난 뒤 결과 값으로 500번지 ~ 505번지에 어떤 값이 저장되는가? 16진수로 나타내시오. [각 0.5점]

500번지	0	503번지	1
501번지	4	504번지	3
502번지	B400	505번지	9800