

학부		학번		반		이름	
----	--	----	--	---	--	----	--

1. Mano의 Basic Computer에서는 인터럽트 서비스가 실행되는 동안 또 다른 인터럽트를 받을 수 없다. 이를 해결하기 위하여 인터럽트 서비스 이후에 돌아갈 리턴 주소와 AC 및 E 레지스터의 내용을 스택에 저장하는 구조로 변경할 때, 다음 물음에 답하라. 스택은 데이터의 마지막에 저장된 데이터가 먼저 나오는 방식의 데이터구조이다. 이 때 저장될 위치를 알려주는 값을 스택포인터가 가지고 있다.

가. 스택은 메모리의 4001번지부터 시작되고,

나. 스택 포인터는 메모리 4000번지에 저장된다.

다. 인터럽트 서비스 루틴이 호출되는 경우, 일단 메모리 0번지에 리턴 주소를 저장하고, 스택에는 Flag E와 AC 값 및 리턴 주소가 저장된다. 돌아올 때는 스택의 Flag E와 AC 값을 복원하고, 리턴주소는 메모리 0번지에 저장하여서 돌아오게 된다. 다음의 인터럽트 서비스 루틴의 빈 곳을 채워라. [각 2점]

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

2. 아래의 프로그램이 수행되는 과정에 대한 물음에 답하여라.[각2점]

LINE	프로그램			LINE	프로그램		
1		ORG	1000	21		STA	Y
2		LDA	N	22		SZE	
3		BSA	MUL	23		BUN	ONE
4	FST,	HEX	0003	24		BUN	ZRO
5	SND,	HEX	0005	25	ONE,	LDA	X
6		HEX	0	26		ADD	P
7		HLT		27		STA	P
8	N,	DEC	-8	28	ZRO,	LDA	X
9		ORG	2000	29		CIL	
10	MUL,	HEX	0	30		STA	X
11		STA	CTR	31		ISZ	CTR
12		LDA	MUL I	32		BUN	LOP
13		STA	X	33		STA	MUL I
14		ISZ	MUL	34		ISZ	MUL
15		LDA	MUL I	35		BUN	MUL I
16		ISZ	MUL	36	CTR,	DEC	0
17		STA	Y	37	X,	HEX	0
18	LOP,	CLE		38	Y,	HEX	0
19		LDA	Y	39	P.	HEX	0
20		CIR					

2.1 프로그램이 끝날 때까지 메모리 2000번지에 저장되는 값의 변화를 작성하라.

0	->	1002	->	1003	->	1004	->	1005
---	----	------	----	------	----	------	----	------

2.2 AC를 통하여 서브루틴과 메인으로 전달되는 값은 각각 얼마인가?

MAIN->SUB	-8	SUB->MAIN	000F
-----------	----	-----------	------

2.3 파라미터로 서브루틴과 메인에 전달되는 값은 각각 몇 개인가?

MAIN->SUB	2	SUB->MAIN	1
-----------	---	-----------	---

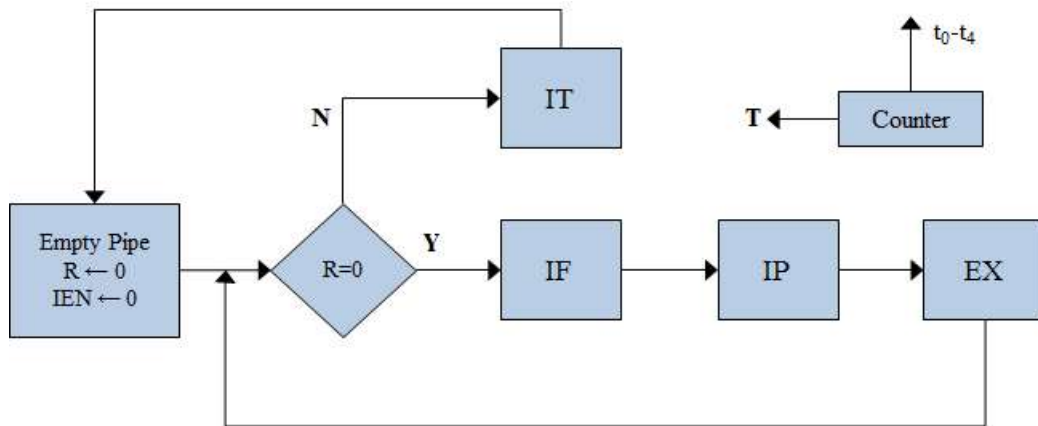
2.4 32번과 33번 명령어가 없는 경우, 동일한 결과를 위해서 메인 프로그램을 어떻게 수정하면 되는가?

1004		STA	RES
1005		HLT	
1006	RES	HEX	0

2.5 하드웨어로 위 알고리즘을 구현한 회로를 무엇이라고 부르는가?

ARRAY MULTIPLIER

3. 다음 그림은 Mano가 설계한 컴퓨터를 위한 instruction pipeline의 제어부분으로 instruction fetch(IF)와 indirect processing(IP), execution(EX)의 세 단계로 구분되고 IF 단계는 interrupt processing(IT)과 정상처리부분(IF)로 구분된다. 전체 pipeline을 동기화하기 위한 time signal T와 각 단계 내부의 time signal $t(t_0 \sim t_4)$ 가 있다고 가정한다. 표는 ISZ instruction을 예로 각 단계를 위한 수정된 RTL이다. [각3점]

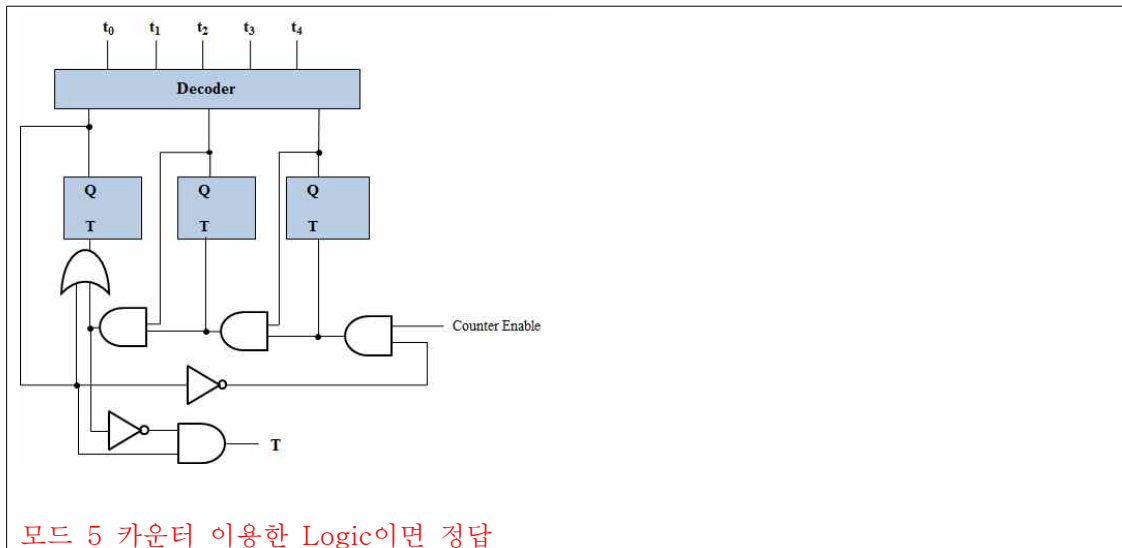


	IF	IT	IP	EX(ISZ)
t_0	No-operation	No-operation	\square : $AR \leftarrow M[AR1]$ \square : $AR \leftarrow AR1$	No-operation
t_1	$AR1 \leftarrow PC$	$AR1 \leftarrow 0$, $TR \leftarrow PC$	No-operation	$DR \leftarrow M[AR]$
t_2	$IR \leftarrow M[AR1]$, $PC \leftarrow PC + 1$	$M[AR1] \leftarrow TR$ $PC \leftarrow 0$	No-operation	$DR \leftarrow DR + 1$
t_3	$D0 \dots D7 \leftarrow \text{dec}(IR(12-14))$, $AR1 \leftarrow IR(0-11)$, $I \leftarrow IR(15)$	$PC \leftarrow PC + 1$	No-operation	$M[AR] \leftarrow DR$, IF($DR=0$) then ($PC \leftarrow PC + 1$)
t_4	No-operation	No-operation	No-operation	

3.1 AR1 register를 따로 두어야 하는 이유를 설명하라.

EX 수행 동안 IF가 일어나기 때문에 IF과정에서 AR1을 이용하고 EX는 AR을 사용한다.

3.2 이 pipeline의 동작을 동기화하기 위한 counter register는 각 단계 내부의 time signal $t_0 \dots t_4$ 를 생성하고, t_4 가 끝날 때마다 생성되는 time signal은 각 단계의 동기화에 이용된다. 이를 위한 counter logic을 설계하라.



3.3 IP 단계의 t_0 에서 각 경우의 RTL을 수행할 조건을 완성하라.

D7'I	$AR \leftarrow M[AR1]$	(D7'I)'	$AR \leftarrow AR1$
------	------------------------	---------	---------------------

3.4 IF, IT, EX 단계의 t_0 에 no-operation이 들어가는 이유를 설명하라.

Data dependency에 대한 언급이나 충돌이 발생하기 때문이라는 언급 시 정답 처리

3.5 EX 단계의 t_4 에 빈칸을 메워라.

(IEN)(FGI+FGO): $R \leftarrow 1$

4. 다음 서브루틴에 대하여 물음에 답하라.[각2점]

109	SH4,	HEX	0
10A		CIL	
10B		CIL	
10C		CIL	
10D		CIL	
10F		AND	MSK
110		BUN	SH4 I
111	MSK,	HEX	FFF0

4.1 AC에 십육진수 1234를 가지고, SH4를 CALL한 후의 결과는 얼마인가?

2340₍₁₆₎

4.2 10F 명령어가 없는 경우에는 얼마인가?

2340₍₁₆₎ (E=0), 2348₍₁₆₎ (E=1)

4.3 4.2의 10F 명령어가 없는 루틴을 SHX라고 할 때, SHX를 활용하여 16비트의 값을 E Flag를 포함하지 않고, 4비트 Circular Shift Left하는 루틴(CSH)을 완성하라.[각2점]

206	CSH,	HEX	0
207		BSA	SHX
208		STA	TMP
209		CIL	
20A		AND	PAD
20B		STA	PAD
20C		LDA	TMP
20D		AND	MSK
20F		ADD	PAD
210		BUN	CSH I
211	PAD	HEX	000F
212	TMP,	HEX	0

5. DSZ라는 memory-reference instruction을 설계하여 ISZ를 대체하려 한다. DSZ는 ISZ와 반대로 주어진 메모리 주소에 있는 값을 1 감소시킨 다음 그 메모리 주소로 다시 저장하고 감소시킨 값이 0이면 다음 instruction을 skip한다. (주의: 명령어 실행 후 본래 AC에 저장되어 있던 값이 변경되어서는 안 된다) [각 5점]

5.1 DSZ instruction을 위한 RTL을 작성하라.

D ₆ T ₄ : $TR \leftarrow AC$	D ₆ T ₄ : $DR \leftarrow M[AR]$
D ₆ T ₅ : $DR \leftarrow M[AR], AC \leftarrow 0$	D ₆ T ₅ : $AC \leftarrow DR, DR \leftarrow AC$
D ₆ T ₆ : $AC \leftarrow \text{complement } AC$	D ₆ T ₆ : $AC \leftarrow \text{complement } AC$
D ₆ T ₇ : $AC \leftarrow AC + DR$	D ₆ T ₇ : $AC \leftarrow AC + 1$
D ₆ T ₈ : $M[AR] \leftarrow AC$, if $(AC = 0)$ then ($PC \leftarrow PC + 1$)	D ₆ T ₈ : $AC \leftarrow \text{complement } AC$
D ₆ T ₉ : $DR \leftarrow TR$	D ₆ T ₉ : $M[AR] \leftarrow AC$, if $(AC = 0)$ then ($PC \leftarrow PC + 1$)
D ₆ T ₁₀ : $AC \leftarrow DR, SC \leftarrow 0$	D ₆ T ₁₀ : $AC \leftarrow DR, SC \leftarrow 0$

5.2 DSZ instruction의 기능을 수행할 수 있는 subroutine을 작성하라.

Line	Subroutine call 방법		
1		BSA	DSZ
2		HEX	(주소값)
※ (주소값)은 특정 메모리의 주소			
Line	Subroutine		
1	DSZ	HEX	0
2		STA	TMP
3		LDA	DSZ I
4		ISZ	DSZ
5		STA	PTR
6		LDA	PTR I

Line	Subroutine		
7		ADD	MIN
8		STA	PTR I
9		SZA	
10		BUN	NSK
11		ISZ	DSZ
12	NSK	LDA	TMP
13		BUN	DSZ I
14	MIN	DEC	-1
15	PTR	HEC	0
16	TMP	HEX	0

6. 다음 프로그램은 원소가 4개인 배열 A와 배열 B를 더하여서 배열 C에 저장하는 서브루틴이다. 배열의 시작주소와 원소의 개수는 파라미터로 서브루틴에 전달된다. 아래 주소의 instruction이 실행된 다음 레지스터와 메모리의 결과를 작성하라. (16진수는 ₍₁₆₎을 표기하시오) [각 2점]

주소				주소			
4		BSA	MAT	6B		ISZ	PT2
5		HEX	100	6C		ISZ	PT3
6		HEX	200	6D		ISZ	CTR
7		HEX	300	6E		BUN	LOP
8		DEC	-4	6F		BUN	MAT I
9		HLT		70	PT1	HEX	0
				71	PT2	HEX	0
		ORG	50	72	PT3	HEX	0
50	MAT,	HEX	0	73	CTR	DEC	0
51		LDA	MAT I			ORG	100
52		STA	PT1	100		DEC	6
53		ISZ	MAT	101		DEC	4
54		LDA	MAT I	102		DEC	7
55		STA	PT2	103		DEC	9
56		ISZ	MAT			ORG	200
57		LDA	MAT I	200		DEC	4
58		STA	PT3	201		DEC	6
59		ISZ	MAT	202		DEC	3
5A		LDA	MAT I	203		DEC	1
5B		STA	CTR			ORG	300
5C		ISZ	MAT	300		DEC	0
5D	LOP,	LDA	PT1 I	301		DEC	0
5E		ADD	PT2 I	302		DEC	0
5F		STA	PT3 I	303		DEC	0
6A		ISZ	PT1				

명령어	4	58	6F
AC		300 ₍₁₆₎	
PC	51 ₍₁₆₎	59 ₍₁₆₎	9 ₍₁₆₎
50번지	5 ₍₁₆₎	7 ₍₁₆₎	9 ₍₁₆₎
72번지		300 ₍₁₆₎	304 ₍₁₆₎
303번지			10

7. MAN0가 설계한 컴퓨터의 Instruction set은 complete하다. 아래의 Instruction 들 중 없어도 instruction completeness가 만족될 수 있는 명령어 세 개를 선택하고, 그 명령어들을 대신할 수 있는 서브루틴을 작성하라.[각 5점]

TABLE 5-6 Control Functions and Microoperations for the Basic Computer

Fetch	$R'T_0$: $AR \leftarrow PC$
	$R'T_1$: $IR \leftarrow M[AR], PC \leftarrow PC + 1$
Decode	$R'T_2$: $D_0, \dots, D_7 \leftarrow \text{Decode } IR(12-14),$ $AR \leftarrow IR(0-11), I \leftarrow IR(15)$
Indirect	D_7IT_3 : $AR \leftarrow M[AR]$
Interrupt:	
	$T_0T_1T_2(IEN)(FGI + FGO)$: $R \leftarrow 1$
	RT_0 : $AR \leftarrow 0, TR \leftarrow PC$
	RT_1 : $M[AR] \leftarrow TR, PC \leftarrow 0$
	RT_2 : $PC \leftarrow PC + 1, IEN \leftarrow 0, R \leftarrow 0, SC \leftarrow 0$
Memory-reference:	
AND	D_0T_4 : $DR \leftarrow M[AR]$ D_0T_5 : $AC \leftarrow AC \wedge DR, SC \leftarrow 0$
ADD	D_1T_4 : $DR \leftarrow M[AR]$ D_1T_5 : $AC \leftarrow AC + DR, E \leftarrow C_{out}, SC \leftarrow 0$
LDA	D_2T_4 : $DR \leftarrow M[AR]$ D_2T_5 : $AC \leftarrow DR, SC \leftarrow 0$
STA	D_3T_4 : $M[AR] \leftarrow AC, SC \leftarrow 0$
BUN	D_4T_4 : $PC \leftarrow AR, SC \leftarrow 0$
BSA	D_5T_4 : $M[AR] \leftarrow PC, AR \leftarrow AR + 1$ D_5T_5 : $PC \leftarrow AR, SC \leftarrow 0$
ISZ	D_6T_4 : $DR \leftarrow M[AR]$ D_6T_5 : $DR \leftarrow DR + 1$ D_6T_6 : $M[AR] \leftarrow DR, \text{ if } (DR = 0) \text{ then } (PC \leftarrow PC + 1), SC \leftarrow 0$
Register-reference:	
	$D_7I'T_3 = r$ (common to all register-reference instructions)
	$IR(i) = B_i$ ($i = 0, 1, 2, \dots, 11$)
	r : $SC \leftarrow 0$
CLA	rB_{11} : $AC \leftarrow 0$
CLE	rB_{10} : $E \leftarrow 0$
CMA	rB_9 : $AC \leftarrow \overline{AC}$
CME	rB_8 : $E \leftarrow \overline{E}$
CIR	rB_7 : $AC \leftarrow \text{shr } AC, AC(15) \leftarrow E, E \leftarrow AC(0)$
CIL	rB_6 : $AC \leftarrow \text{shl } AC, AC(0) \leftarrow E, E \leftarrow AC(15)$
INC	rB_5 : $AC \leftarrow AC + 1$
SPA	rB_4 : If $(AC(15) = 0)$ then $(PC \leftarrow PC + 1)$
SNA	rB_3 : If $(AC(15) = 1)$ then $(PC \leftarrow PC + 1)$
SZA	rB_2 : If $(AC = 0)$ then $PC \leftarrow PC + 1$
SZE	rB_1 : If $(E = 0)$ then $(PC \leftarrow PC + 1)$
HLT	rB_0 : $S \leftarrow 0$
Input-output:	
	$D_7IT_3 = p$ (common to all input-output instructions)
	$IR(i) = B_i$ ($i = 6, 7, 8, 9, 10, 11$)
	p : $SC \leftarrow 0$
INP	pB_{11} : $AC(0-7) \leftarrow INPR, FGI \leftarrow 0$
OUT	pB_{10} : $OUTR \leftarrow AC(0-7), FGO \leftarrow 0$
SKI	pB_9 : If $(FGI = 1)$ then $(PC \leftarrow PC + 1)$
SKO	pB_8 : If $(FGO = 1)$ then $(PC \leftarrow PC + 1)$
ION	pB_7 : $IEN \leftarrow 1$
IOF	pB_6 : $IEN \leftarrow 0$

CLA			CME			INC		
IS1,	HEX	0	IS2,	HEX	0	IS3,	HEX	0
	AND	ZRO		STA	TMP		ADD	ONE
	BUN	IS1 I		CIR			BUN	IS3 I
ZRO	DEC	0		CMA		ONE	DEC	1
				CIL				
				LDA	TMP			
				BUN	IS2 I			
			TMP	HEX	0			

8. 다음 각 설명들에 대하여 틀린 부분을 수정하라.[각 2점]

- (1) RISC컴퓨터의 성능을 충분히 발휘시키려면, 적은 수의 레지스터들을 활용하여 컴파일러가 최적의 코드를 생성하여야 한다.
- (2) Floating point 숫자의 덧셈연산을 pipeline으로 할 때, normalization이 필요한 이유는 mantissa field의 범위가 $0.1 < \text{mantissa} < 1$ 을 유지하여야 하기 때문이다.
- (3) Pipelining 처리에서 실제의 성능이 이상적인 성능인 Speedup이 세그먼트의 수만큼 나오지 못하는 이유 중 하나는 세그먼트의 실행 시간이 일정하기 때문이다.
- (4) 10 MIPS의 CPU와 10 CPS의 I/O장치 간에 입출력을 polling으로 수행하는 경우 발생하는 waste time은 1초이다.
- (5) DMA 전송에서 메모리와 CPU사이에 BUS의 연결은 BUS request가 오기 전에 high impedance 상태로 연결을 단절하여야 한다.

1	적은 수 -> 많은 수
2	$0.1 < \text{mantissa} < 1$ -> $0.5 \leq \text{mantissa} < 1$
3	세그먼트의 실행시간이 일정하기 때문이다. -> Setup time 또는 Pipeline conflict 때문이다.
4	1초 -> 0.1초
5	BUS Request가 오기 전에 -> BUS Grant를 하기 전에