2013년 1학기 컴퓨터구조 기말고사

학과		학번		이름	

1. Interrupt nesting can not be occurred in Mano's Basic Computer. In order to solve this problem, the return address and global values should be stored in the stack. We would make an interrupt service routine which includes stack and its operations. The stack has a stack pointer which indicates top position.

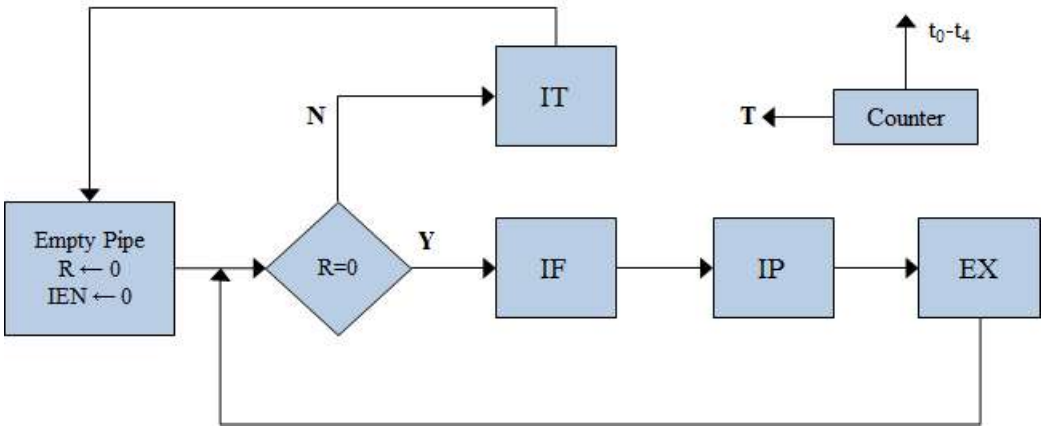
1) Stack starts at the location 4001 in memory.

2) Stack pointer locates 4000 in memory.

3) When an interrupt occurs, the return address is stored at the location 0(zero) in memory. Fill the blanks. [1 point each]

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

2. We would extend Mano's basic computer to have instruction pipeline. It has three consecutive steps - instruction fetch(IF), indirect processing(IP), and execution(EX) - in its instruction cycle. IF step is switched to Interrupt step(IT) as soon as interrupt occurs. We have time signal T and internal time signal t(t_0~t_4) at each steps, in order to synchronize pipeline. Table shows modified RTL for the ISZ instruction.[16 points]



	IF	IT	IP	EX(ISZ)
t_0	No-operation	No-operation	□: AR ← M[AR1] □: AR ← AR1	No-operation
t_1	AR1 ← PC	AR1 ← 0, TR ← PC	No-operation	DR ← M[AR]
t_2	IR ← M[AR1], PC ← PC + 1	M[AR1] ← TR PC ← 0	No-operation	DR ← DR + 1
t_3	D0…D7 ← dec(IR(12-14)), AR1 ← IR(0-11), I ← IR(15)	PC ← PC + 1	No-operation	M[AR] ← DR, IF(DR=0) then (PC ← PC + 1)
t_4	No-operation	No-operation	No-operation	

2.1 Explain why we should have additional AR1 register? [3 points]

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

2.2 Fill the RTL at time t_0 of IP step.[2 points each]

D7'I	AR← M[AR1]	(D7'I)'	AR← AR1

2.3 What is the reason no-operation is required at time t_0 IF, IT, EX steps. [3 points]

Data dependency 혹은 충돌이 발생하기 때문이라는 언급 시 정답처리

2.4 Fill the blank at time t_4 of EX steps.[3 points]

(IEN)(FGI+FGO): R ← 1

2.5 What is the minimum size of program to get a pipeline effect of speedup? Answer the number of instructions. [3 points]

5

3. Decide following statements whether it represents the characteristic of CISC or RISC. [1 point each]

Characteristic	C or R
Most instructions are used equally	R
A large variety of address modes	C
Many memory reference instructions in instruction set	C
Single-cycle instruction execution	R
Micro-programed control is more preferable.	C
Intelligent compiler should be required to get an efficiency	R
Overlapped window is useful to reduce procedure call burden	R
Instruction pipeline should be used	R
A relatively large number of registers	R
the number of operand can be from zero to three.	C

4. Answer to the following questions.[17 points]

LINE	프로그램			LINE	프로그램		
1		ORG	2000	21		STA	Y
2		LDA	N	22		SZE	
3		BSA	MUL	23		BUN	ONE
4	FST,	DEC	0002	24		BUN	ZRO
5	SND,	DEC	0003	25	ONE,	LDA	X
6	TRD,	HEX	0	26		ADD	P
7		HLT		27		STA	P
8	N,	DEC	-2	28	ZRO,	CLE	
9		ORG	3000	29		LDA	X
10	MUL,	HEX	0	30		CIL	
11		STA	CTR	31		STA	X
12		LDA	MUL I	32		ISZ	CTR
13		STA	X	33		BUN	LOP
14		ISZ	MUL	34		STA	MUL I
15		LDA	MUL I	35		ISZ	MUL
16		STA	Y	36		BUN	MUL I
17		ISZ	MUL	37	CTR,	DEC	0
18	LOP,	CLE		38	X,	HEX	0
19		LDA	Y	39	Y,	HEX	0
20		CIR		40	P.	HEX	0

4.1 Fill up the blanks which are the values being changed during program execution. [1 point each]

Y	0	->	3	->	1	->	0

X	0	->	2	->	4	->	8

4.2 What are the global values will be transferred from main to subroutine?[1 point each]

-2 (AC)	0 or 1 (E)

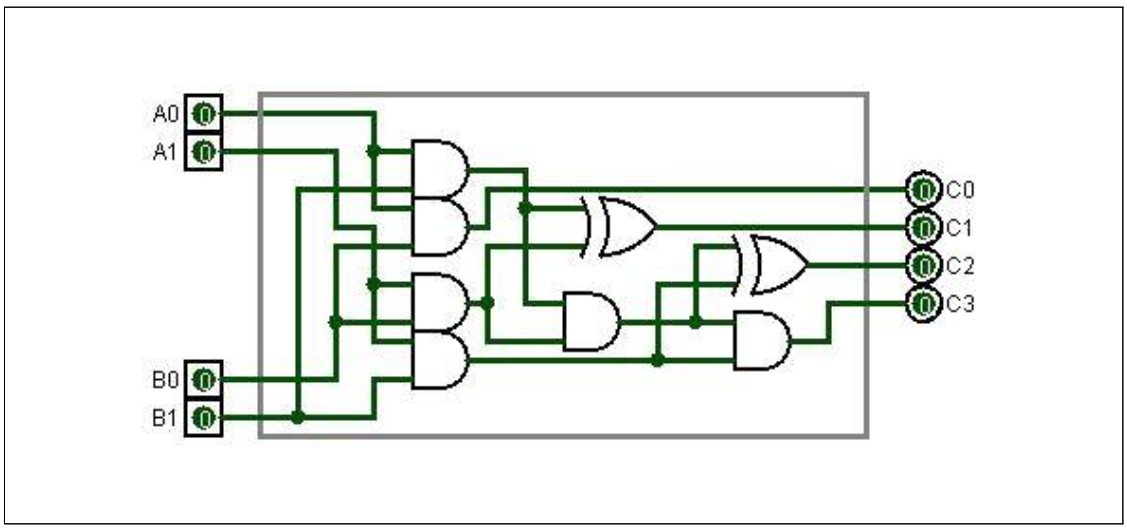
4.3 In what location the final results are stored? Answer the exact address of it.[2 points]

2004

4.4 In order to get a same results without two instruction 34 and 35, we have to insert an instruction between instruction 6 and 7.[2 points]

LINE	프로그램		
		STA	TRD

4.5 We can design an array multiplier circuit for the same function with this subroutine MUL. Draw a logic circuit of 2-bit array multiplier [5 points]



5. We would like to replace a MRI-type DSZ instruction with ISZ instruction. DSZ instruction decrease the value of operand by1(one) and store decreased value at the same address. If that value is 0(zero), increase PC by 1. (Alret: The Value of AC should not be changed, after the execution of DSZ)[10 points]

5.1. Write RTL for DSZ instruction.[5 points]

D_6T_4: TR ← AC	D_6T_4: DR ← M[AR]
D_6T_5: DR ← M[AR], AC ← 0	D_6T_5: AC ← DR, DR ← AC
D_6T_6: AC ← complement AC	D_6T_6: AC ← complement AC
D_6T_7: AC ← AC + DR	D_6T_7: AC ← AC + 1
D_6T_8: M[AR] ← AC, if (AC = 0) then (PC ← PC + 1)	D_6T_8: AC ← complement AC
D_6T_9: DR ← TR	D_6T_9: M[AR] ← AC, if (AC = 0) then (PC ← PC + 1)
D_6T_{10}: AC ← DR, SC ← 0	D_6T_{10}: AC ← DR, SC ← 0

5-2. Program a subroutine whose operation is the same as DSZ instruction.[5 points]

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. A task is consist of k segments. And we assume t_p as the execution time of one segment and t_s as the execution time of one tasks. When we execute n tasks on the k-segment pipeline, answer to the following questions.[2 points each]

6-1. Derive an equation of speedup(S) of pipeline processing.

S = $\frac{nt_s}{(k+n-1)t_p}$

6-2. List relationship between n and k, to achieve S more than 1(one).

$n > k$

6-3. Pipeline should have something between processing unit of segments. One of the reasons is the difference between the execution time of each segment. What is it?

register 또는 buffer

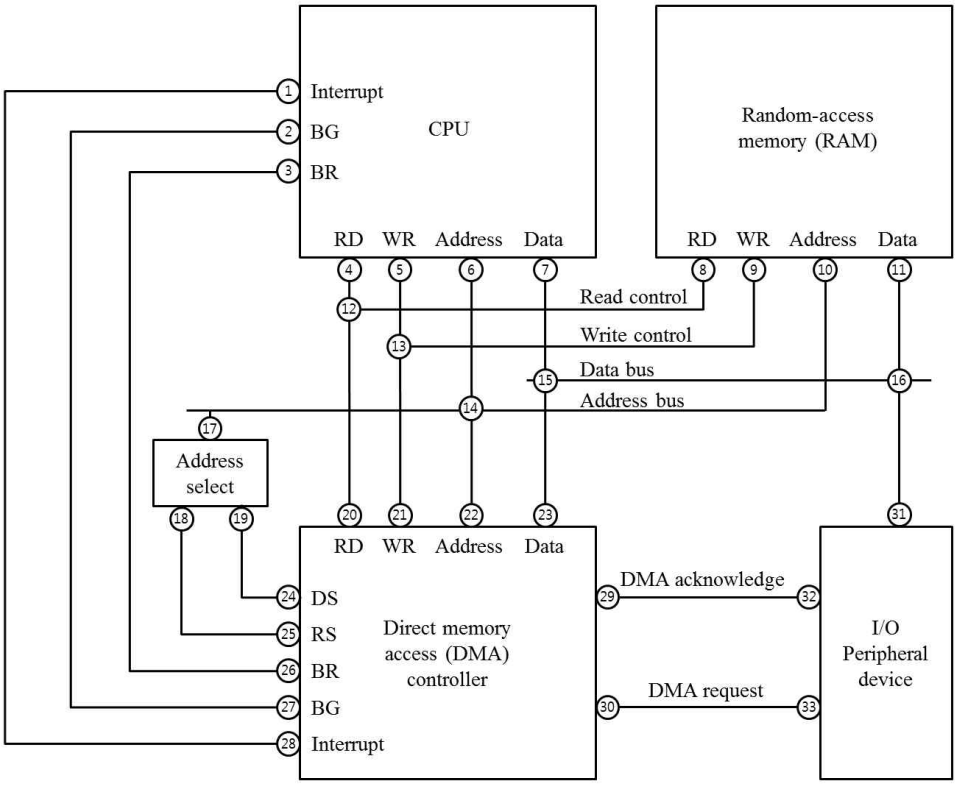
6-4. We could not achieve ideal speedup of instruction pipeline processing practically, List three obstacles of it.

Resource conflicts

Data dependency

Branch difficulties, 그 외에 세그먼트의 크기가 서로 다르다는 내용도 정답 처리

7. When your answer is a path in the figure, write start number and ending number(ex, 19->24) as a path.[20 points]



7.1 Which path represents the final step of DMA operation?[2 points] When is the final step triggered?[2 points]

28→1
word count register의 값이 0이 되었을 때

7.2 List all numbers which have both directions from number 12 to 16. Explain the reason why they have both directions.[2 points]

12, 13, 14, 15, 16 [1점] 평소에는 CPU가 메모리에 접근하기 위해 버스를 제어하며, DMA transfer 시에는 DMA가 I/O device와 메모리 간에 직접 데이터를 전송하기 위해 CPU로부터 bus control 권한을 가져온다. 따라서 이 두 경우를 위해 양방향으로 연결이 되어야 한다. [1점]

7-3 List all numbers which should have direction arrow from 24 to 33.[2 points]

24, 25, 27, 30, 31, 32

7-4. When CPU sends DMA controller the address of memory data read or written, which path is used?[2 points]

DMA가 CPU에 의해 초기화 될 때, 메모리 블록의 시작 주소 또는 데이터가 저장된 주소가 data bus를 통해 전송된다. [1점] 7→23 or 7→15→23 [1점]

7-5. When DMA controller transfers the address of memory data read or written to memory, which path is used?[2 points]

22→10 or 22→14→10

7-6. DMA controller has three registers. What are they?[2 points each]

Address register	Word count register	Control register

7-7. What is a path to transfer data using DMA operation?[2 points]

11↔31