

1. 다음의 프로그램에 대하여 빈칸을 채우고 기계어 코드로 변환하시오.

어셈블리 언어 프로그램	메모리 주소	기계어 (2진수)							기계어 (16진수)
		OP	ALU	CC	DR	SR	V	SR2/VAL	
.ORIGIN 0x2000 LOAD R0, x CMP R0, 10 BR n, L2 SUB R0, R0, 10 L2: STORE R0, y SWI 255 x: .FILL 17 y: .BLOCK 1	0x2000								
	0x2001								
	0x2002								
	0x2003								
	0x2004								
	0x2005								
	0x2006								
	0x2007								

2. 다음의 프로그램에 대하여
(a) 빈칸을 채우고 어셈블리 언어로 변환하시오.

메모리 주소	기계어 (16진수)	기계어 (2진수)							어셈블리 언어
		OP	ALU	CC	DR	SR	V	SR2/VAL	
0x2000	0x97f10000								
0x2001	0x07810041								
0x2002	0x03810030								
0x2003	0x97f10001								
0x2004	0x63fffffb								

(b) 어떤 목적의 프로그램인가?

(참고) 아스키 코드: ‘A’ 는 0x41, ‘0’ 는 0x30

(참고) *TOY* 프로세서 기계어 코드 표

	b ₃₁	b ₂₉	b ₂₈	b ₂₆	b ₂₅	b ₂₃	b ₂₂	b ₂₀	b ₁₉	b ₁₇	b ₁₆	b ₁₅	b ₁₃	b ₁₂	b ₀
	OP	ALU	CC	DR	SR	V	SR2	VAL							
ADD	0 0 0	0 0 0	1 1 1	Rd	Rs	V									
SUB	0 0 0	0 0 1	1 1 1	Rd	Rs	V									
CMP	0 0 0	0 0 1	0 0 0	0 0 0	Rs	V									
AND	0 0 0	0 1 0	1 1 1	Rd	Rs	V									
OR	0 0 0	0 1 1	1 1 1	Rd	Rs	V									
XOR	0 0 0	1 0 0	1 1 1	Rd	Rs	V									
NOT	0 0 0	1 0 0	1 1 1	Rd	Rs	1	1 1 1	1 1 1 1							
COPY	0 0 0	1 0 1	1 1 1	Rd	0 0 0	V									
LSL	0 0 0	1 1 0	1 1 1	Rd	Rs	1	0 0 0	0 n							
LSR	0 0 0	1 1 0	1 1 1	Rd	Rs	1	0 0 0	1 n							
ASL	0 0 0	1 1 1	1 1 1	Rd	Rs	1	0 0 0	0 n							
ASR	0 0 0	1 1 1	1 1 1	Rd	Rs	1	0 0 0	1 n							
LOAD	0 0 1	0 0 0	1 1 1	Rd	1 1 1	1									
LDR	0 0 1	0 0 0	1 1 1	Rd	Ra	1									
STORE	0 1 0	0 0 0	0 0 0	Rs	1 1 1	1									
STR	0 1 0	0 0 0	0 0 0	Rs	Ra	1									
BR	0 1 1	0 0 0	N Z P	1 1 1	1 1 1	1									
BRR	0 1 1	0 0 0	N Z P	1 1 1	Ra	1									
LEA	0 1 1	0 0 0	1 1 1	Rd	1 1 1	1									
LINK	0 1 1	0 0 0	1 1 1	1 1 0	1 1 1	1	0 0 0	0 0 0 1							
RET	0 1 1	0 0 0	1 1 1	1 1 1	1 1 0	1	0 0 0	0 0 0 0							
SWI	1 0 0	1 0 1	1 1 1	1 1 1	0 0 0	1	0 0 0	n							
RTI	1 0 0	0 0 0	1 1 1	1 1 1	1 1 0	1	0 0 0	0 0 0 0							

(뒷면 계속)

3. 다음을 설명하시오.

스택 포인터의 용도	
다음의 TOY 어셈블리 언어 표현 (x, y, z는 전역변수임) x = y + (z - 2);	
TOY 어셈블리 프로그램에서 함수 호출 방법과 운영체제 기능 호출 방법의 차이	
함수 호출시 파라미터 전달에 레지스터를 이용하는 방법과 스택을 이용하는 방법의 장단점 비교	
장치 제어기의 상태/제어 레지스터	

4. 다음은 TOY C 컴파일러의 출력이다. 부분별로 내용을 설명하시오

main: ... LOAD R0, y COPY R1, 0 CMP R0, R1 BR nz, _L2 LOAD R0, y STORE R0, x BR nzp, _L3 _L2: SUB R5, R5, 1 LOAD R0, y STR R0, R5, 0 LINK BR nzp, f ADD R5, R5, 1 STORE R0, x	이 부분의 원래의 C 언어 표현은?
_L3: ... f: SUB R5, R5, 2 STR R6, R5, 1 STR R4, R5, 0 COPY R4, R5	이 부분의 목적은?
LDR R0, R4, 2 NOT R0, R0 ADD R0, R0, 1	함수 f를 원래의 C 언어로 표현하면?
_L4: COPY R5, R4 LDR R4, R5, 0 LDR R6, R5, 1 ADD R5, R5, 2 RET	이 부분의 목적은?
_DATA: x: .BLOCK 1 y: .FILL 7 _END	이 부분의 원래의 C 언어 표현은?