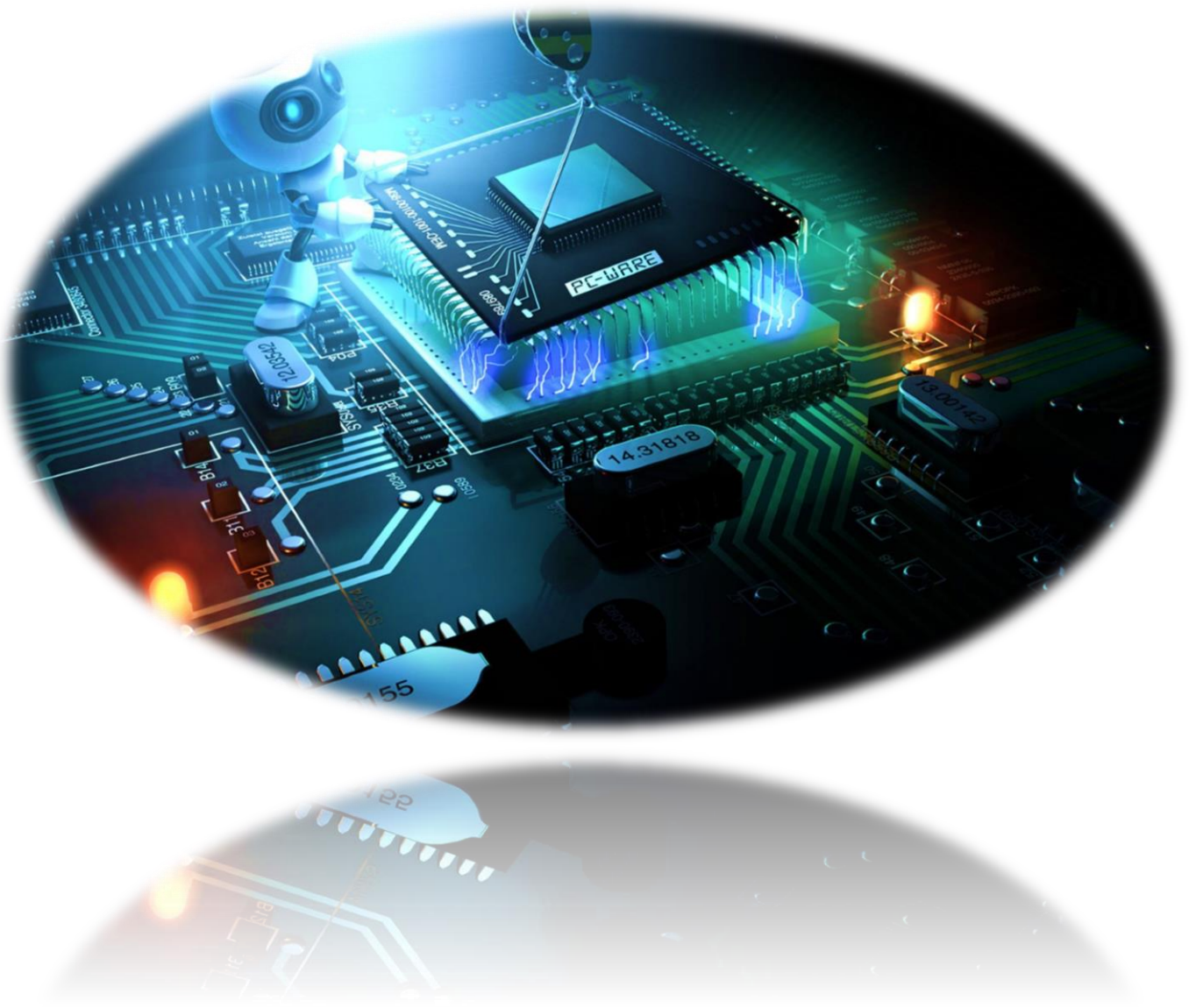

EJERCICIOS SEMANA 2-3

ESTRUCTURA DE COMPUTADORES



BRYAN MORENO PICAMÁN

Contenido

Descripción..... 2

3.1 Solución 3

3.2 Solución 3

3.3 Solución 3

3.4 Solución 3

3.5 Solución 4

3.30 Solución 4

3.46 Solución 4

3.47 Solución 4

Descripción

Cap.3 CS: APP (Bryant/O'Hallaron)

Probl. 3.1-3.5 pp. 204, 208, 210-211

Probl. 3.30 p. 257

Probl. 3.46-3.47 pp. 305, 310

3.1 Solución

Operando	Valor
%eax	0x100
0x104	0xAB
\$0X108	0X108
(%eax)	0xFF
4(%ecx,%edx)	0xAB
9(%ecx,%edx)	0x11
260(%ecx,%edx)	0x13
0xFC(,%ecx,4)	0xFF
(%eac,%edx,4)	0x11

3.2 Solución

movl	%eax, (%esp)
movw	(%eax)%dx
movb	\$0xFF,%bl
movb	(%esp,%edx,4),%dh
pushl	\$0xFF
movw	%dx, (%eax)
popl	%edi

3.3 Solución

movb	\$0xF, (%bl)
movl	%ax, (%esp)
movw	(%eax), 4(%esp)
movb	%ah, %sh
movl	%eax, \$0x123
movl	%eax, %dx
movb	%si, 8(%ebp)

3.4 Solución

src_t	dest_t	Instrucción
int	int	movl %eax, (%edx)
char	int	movsbl %al, (%edx)
char	unsigned	movsbl %al, (%edx)
unsigned char	Int	movzbl %al, (%edx)
Int	Char	movb %al, (%edx)
unsigned	unsigned char	movb %al, (%edx)
unsigned	int	movl %al, (%edx)

3.5 Solución

xp en %ebp+8,	yp en %ebp+12,	zp en %ebp+16
movl	8(%ebp), %edi	get xp
movl	12(%ebp), %edx	get yp
movl	16(%ebp), %ecx	get zp
movl	(%edx), %ebx	get y
movl	(%ecx), %esi	get z
movl	(%edi), %eax	get x
movl	%eax, (%edx)	Store x at yp
movl	%ebx, (%ecx)	Store y at zp
movl	%esi, (%edi)	Store z at xp

3.30 Solución

- a.- %eax se pone al valor de la dirección de la instrucción popl
- b.- No es una llamada a procedimiento, el control sigue el mismo orden que la instrucción y la dirección retornada es sacada del stack
- c.- Es la forma de tener el valor del contador de programa en un registro entero.

3.46 Solución

- a.- Si seguimos esta regla unos 25 años
- b.- 53 años
- c.- Esto haría que acortáramos unos 6 años el tiempo esperado para alcanzar estas cifras.

3.47 Solución

src_t	dest_t	Instrucción	S	D
long	long	movq	%rdi	%rax
int	long	movslq	%edi	%rax
char	long	movsbq	%dil	%rax
unsigned int	unsigned long	movl	%edi	%eax
unsigned char	unsigned long	movzbq	%dil	%rax
unsigned char	unsigned long	movzbl	%dil	%eax
long	int	movslq	%edi	%rax
long	int	movl	%edi	%eax
unsigned long	unsigned	movl	%edi	%eax