

Examen-Enero-2020-AC-ResueltoCor...



CodeWolf



Arquitectura de Computadores



2º Grado en Ingeniería Informática



**Escuela Politécnica Superior de Córdoba
Universidad de Córdoba**



[Accede al documento original](#)



Escuela de
Organización
Industrial

Contigo que evolucionas.
Contigo que lideras. Contigo que transformas.

**Esto es EOI.
Mismo propósito,
nueva energía.**



Descubre más aquí



EOI Escuela de
Organización
Industrial

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo espacio



Examen Enero 2020

1) En una memoria segmentada, que es el descriptor de segmentos y su estructura.

★ RESOLUCIÓN:

Descriptor de segmento: Contienen la dirección base y el límite del segmento en el espacio lógico, su estructura es la siguiente:

Estructura: Dirección base | Longitud | Protección

→ Dirección: Dirección de la tabla de Páginas / Dirección del Bloque Base.

→ Longitud: Tamaño del segmento.

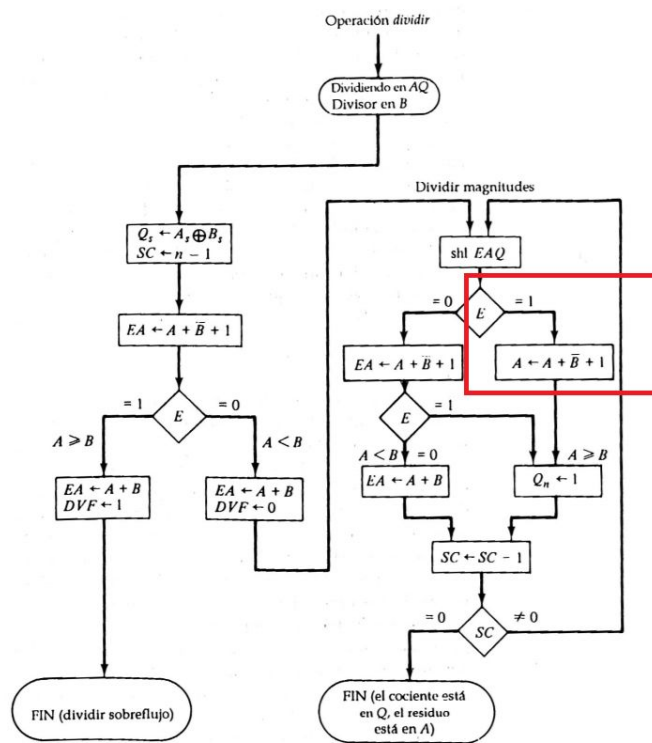
→ Protección: Permisos de acceso a dicho Segmento

◆ Para Mapeo de Página Segmentada, cada Página tendrá una propia

◆ Derechos típicos:

- Privilegios de lectura y escritura completa
- Sólo lectura (protección contra escritura)
- Ejecución (protección de programa)
- Sistema solamente (protección de Sistema Operativo)

2) Que solución da la sección marcada.



★ **RESOLUCIÓN:**

Al ser $F = 1$ se puede restar directamente debido a que $F \mid A$ es $> B$.

- 3) Se tiene un vector A cuya dirección base se encuentra en el registro \$s0 y cuyo número de elementos N se encuentra en el registro \$s1. Realizar un programa mediante instrucciones MIPS que calcule el valor mayor y menor de los elementos de A, dejando dichos valores en los registros \$s2 y \$s3 respectivamente.

★ **RESOLUCIÓN:**

Código Planteado en C :

```
for (int i = 0; i < N ; i++) {  
    if (max < v[i]) {  
        max = v[i];  
    }  
    if (v[i] < min) {  
        min = v[i];  
    }  
}
```

❖ **Código MIPS:**

```
add $s2, $zero, $zero;  
lw $s2, 0 ($s1);  
lw $s3, 0 ($s1);  
loop sll $t1,$t0, 2;  
add $t1 , $t1,$s1;  
lw $t1, 0 ($t1);  
if slt $t2, $s2 , $t1;  
bne $t2 , $zero else ;  
add $s2 , $t1, $zero;  
j FIN_LOOP;  
else slt $t2, $t1 , $s3;  
bne $t3 , $zero FIN_LOOP ;  
add $s3 , $t1, $zero;  
FIN_LOOP:  
addi $t0 , $t0 , 1;  
slt $t1, $t0 , $s1 ;  
bne $s1 , $zero , loop;  
FIN
```

4) En un sistema de memoria virtual paginado el espacio de direcciones es de 256KB, mientras que el espacio de memoria es de 128KB. El tamaño de página es de 32KB. Indicar:

a. Formato de la dirección virtual y formato de la dirección física. Imagínese que se comienza con la memoria principal vacía y que el algoritmo de sustitución que posee es el LRU. Si se llaman a las siguientes páginas en el orden:

7,0,3,0,7,2,0,4,5

¿Cuál será el contenido de la tabla de páginas después de la última llamada?

A partir de la tabla de páginas anteriores, determinar los rangos de direcciones virtuales que producirán un fallo de página en caso de ser accedidas por la CPU.

Dirección física que se corresponde con las siguientes direcciones virtuales del proceso, en caso de que la traducción sea posible:

- 2589D
- 291D7

RESOLUCIÓN:

M.V => 256K => $2^8 * 2^{10} = 2^{18}$ => 18 bits para direccionar M.V

M.V => 128K => $2^7 * 2^{10} = 2^{17}$ => 17 bits para direccionar M.P

M.V => 32K => $2^5 * 2^{10} = 2^{15}$ => 15 bits para direccionar Palabra

FORMATO DE DIRECCIÓN DE MV

Página	Desplazamiento/ Palabra
3	15

FORMATO DE DIRECCIÓN DE MP

Bloque	Desplazamiento/ Palabra
2	15

Bloque	Página
0	7,5
1	0
2	3,4
3	2

Quedando las páginas 5, 0, 4 y 2.

Direcciones físicas de :

2589D

291D7

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo espacio



Necesito concentración

ali ali ooh
esto con 1 coin me
lo quito yo...

WUOLAH

Dirección Hex	Página	Desplazamiento
2589D	10 0	101 1000 1001 1101
	Página 4	
	Página => 4 => Bloque 3 =>	
1D89D	1 1	101 1000 1001 1101
291D7	10 1	001 0001 1101 0111
	Página 5 => Bloque 0	
11D7	0 0	000 0001 1101 0111

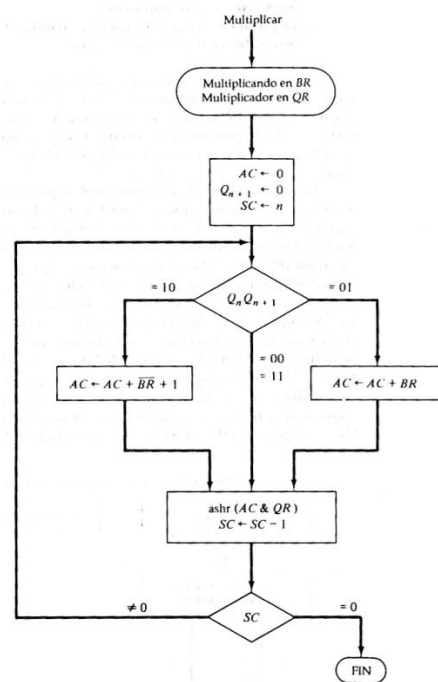
5) Consideremos que, a la estructura de la computadora mejorada mostrada en la figura, se le añade:

- Controlador SC en la unidad de control para la realización de bucles, con su bit asociado ZSC.
- Registro QR: conectado con la memoria principal y con el registro GPR. A su vez conectado con el ACC, de modo que permiten desplazamientos a la derecha e izquierda con el conjunto formado por F-ACC-QR.

Implemente la instrucción MLTB, encargada de realizar la multiplicación en complemento a dos, utilizando el algoritmo de Booth, del contenido del registro QR por el contenido de la posición de memoria "m" (que deberá colocarse en el registro GPR). Realícelo:

- a) Mediante control microprogramado (con señal de enable en la LCB).
- b) Mediante Control Cableado.

WUOLAH



Multiplicación en complemento a dos (Algoritmo de Booth)

★ RESOLUCIÓN:

Microprogramado

MC2 + 0	M=>GPR	0	0	0	-----
MC2 + 1	0 => ACC ; 0 => Qn+1 ; N => SC	0	0	0	-----
MC2 + 2	¿Qn * Qn+1? SC -1 -> SC	0	1	1	MC2 + 6
MC2 + 3	¿Qn?	1	0	0	MC2 + 5
MC2 + 4	ACC + GPR => ACC	0	0	1	MC2 + 6
MC2 + 5	ACC + NOT (GPR) + 1 => ACC	0	0	0	-----
MC2 + 6	ashr (ACC & QR) ;	1	0	1	MC2 + 2
MC2 + 7	-----	0	0	1	FETCH + 0

Imagínate aprobando el examen

Necesitas tiempo y concentración

Planes	 PLAN TURBO	 PLAN PRO	 PLAN PRO+
 Descargas sin publi al mes	10 	40 	80 
 Elimina el video entre descargas			
 Descarga carpetas			
 Descarga archivos grandes			
 Visualiza apuntes online sin publi			
 Elimina toda la publi web			
 Precios Anual <input type="checkbox"/>	0,99 € / mes	3,99 € / mes	7,99 € / mes

Ahora que puedes conseguirlo,
¿Qué nota vas a sacar?



WUOLAH

LCB									
S2	S1	S0	Qn	Qn+1	Zsc		I	B	R
0	0	0	X	X	X		1	0	0
0	0	1	X	X	X		0	1	0
0	1	0	X	X	X		0	0	1
0	1	1	0	0	X		0	1	0
0	1	1	0	1	X		1	0	0
0	1	1	1	0	X		1	0	0
0	1	1	1	1	X		0	1	0
1	0	0	0	X	X		1	0	0
1	0	0	1	X	X		0	1	0
1	0	1	X	X	0		0	1	0
1	0	1	X	X	1		1	0	0

➤ Cableado

MC2 + 0	q0 * t3	M => GPR	SR + 1 => SR
	q0 * t4	0 => ACC ; 0 => Qn+1 ; N => SC	SR + 1 => SR
	q0 * t5 * Qn * NOT(Qn+1)	ACC + NOT (GPR) + 1 => ACC	SR + 1 => SR
	q0 * t5 * NOT (Qn) * Qn+1	ACC + GPR => ACC	SR + 1 => SR
	q0 * t5	-----	SR + 1 => SR
	q0 * t6	ashr (ACC & QR) ; SC -1 -> SC	SR + 1 => SR
	q0 * t7 * Zsc	-----	0 => SR
	q0 * t7 * NOT(Zsc)	-----	5 => SR