Examen Enero 2018 RESUELTO (Revi...

TEAM_GETPPID__

Arquitectura de Computadores

2º Grado en Ingeniería Informática

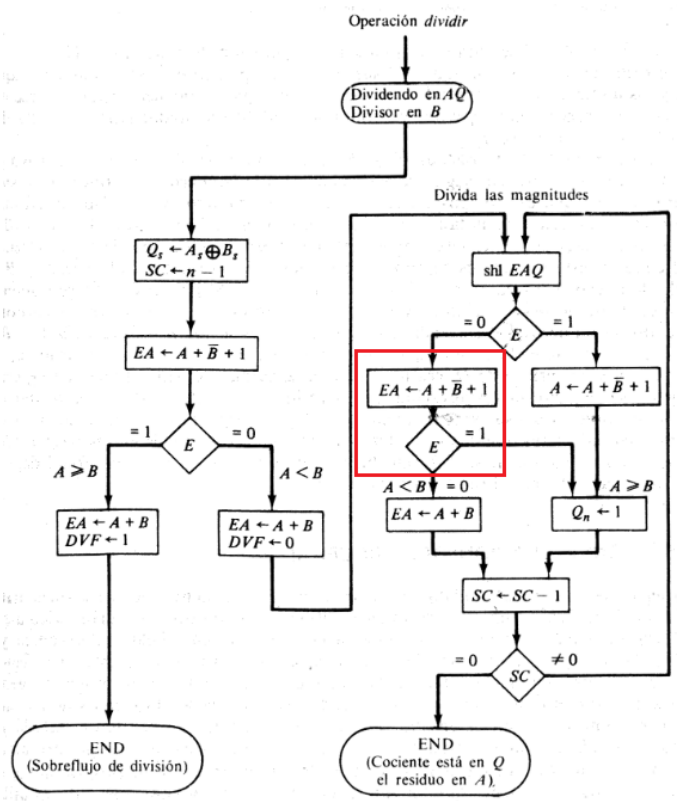
Escuela Politécnica Superior de Córdoba
Universidad de Córdoba

Accede al documento original

Resuelto por TeamGetppid()

Examen enero 2018

1) Dibujar el esquema de una unidad de control microprogramada. A su vez, explicar brevemente la función de cada uno de los bloques.
2) Algoritmo de la división en Signo-Magnitud. Explicar que problema soluciona la sección marcada.



3) Se tiene un vector A cuya dirección base se encuentra en el registro $s0 y cuyo numero 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.
4) En un sistema de memoria virtual paginado el espacio de direcciones es de 128KB, mientras que el espacio de memoria es de 32KB. El tamaño de página es de 8KB. 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:

        13,8,9,8,4,13,2,8,9

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

b. 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.
c. Dirección física que se corresponde con las siguientes direcciones virtuales del proceso, en caso de que la traducción sea posible:
    a) 03A8F
    b) 13A8F

5) Consideremos que, a la estructura de la computadora mejorada mostrada en la figura, se le añade:
    a. Controlador SC en la unidad de control para la realización de bucles, con su bit asociado ZSC.
    b. 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.

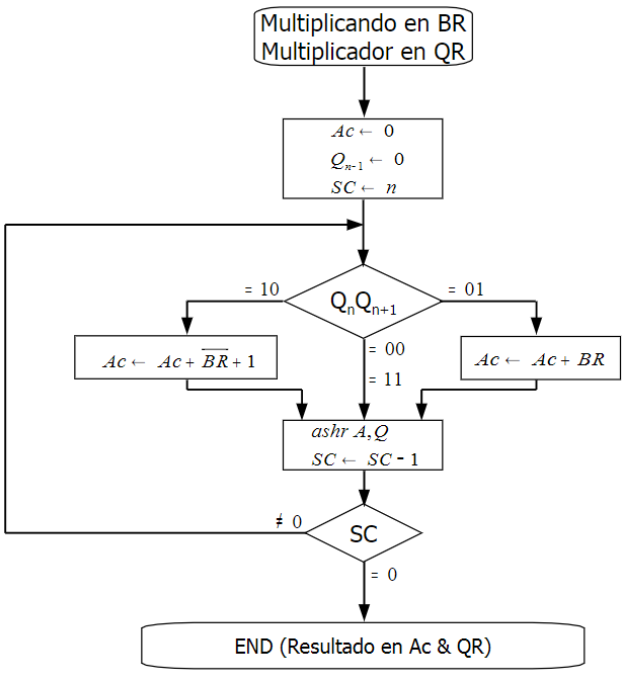
    Implemente la instrucción MLTB, encargada de realizar la multiplicación en complemento a dos, utilizando el algoritmo de Booth, del contenido del registro QR por el contenido de la posición de memoria "m" (que deberá colocarse en el registro GPR). Realícelo:

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

    Incluir el ciclo de búsqueda.

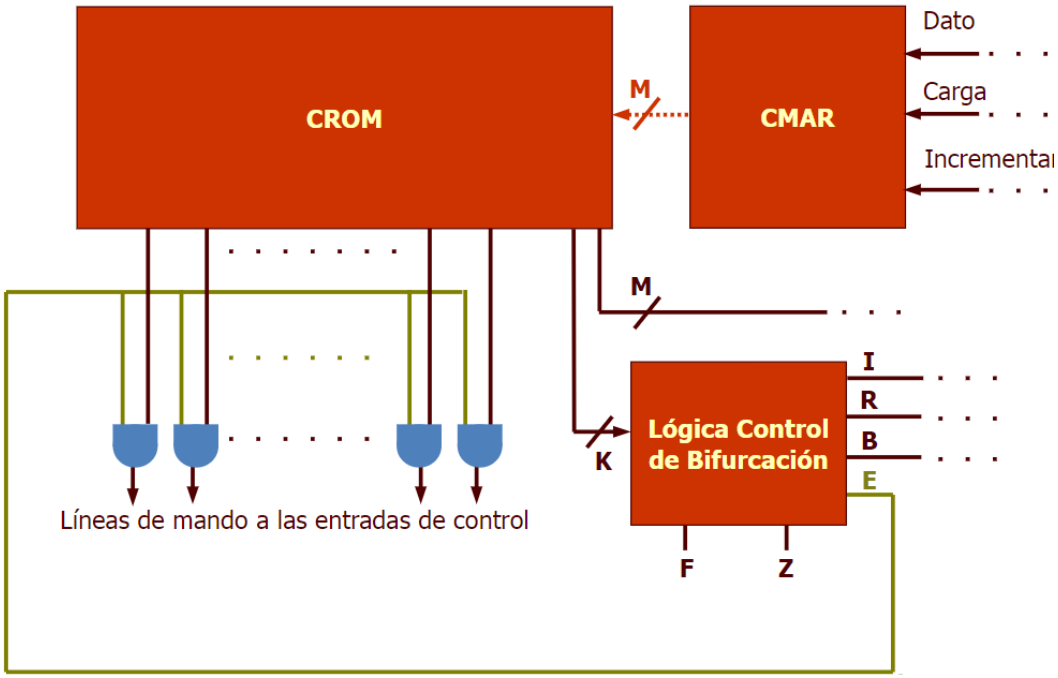
NOTA: EN ESTOS ALGORITMOS PREGUNTAR AL PROFESOR SI HAY QUE BAJAR O NO EL DATO.

EN CASO DE QUE HAYA QUE BAJARLO HAY QUE INCLUIR LA INSTRUCCIÓN (M → GPR) O DE LO CONTRARIO EL EJERCICIO ESTA COMPLETO MAL.



SOLUCIONES

1) Se ha propuesto lo siguiente:



CMAR: El secuenciamiento se consigue variando la dirección contenida en el registro de dirección de acceso a la CROM.
Incremento: Pasar al microinstrucción que se encuentra en la dirección CROM consecutiva => Incrementar el contenido de CMAR.
Bifurcación: Saltar a una microinstrucción que se encuentra en una dirección CROM en particular → Dicha dirección debe ser indicada (codificación de la dirección en la propia micro palabra) → Carga paralela de dicha dirección.
Carga de rutina: Saltar a la microinstrucción de una de las rutinas de las instrucciones → La dirección de inicio de cada instrucción debe conocerse → Carga paralela de dicha dirección
CROM: Contiene las microinstrucciones que serán ejecutadas.

2) Se comprueba mediante suma en C2 del registro Acumulador y GPR si el número cabe en el divisor, para ello posteriormente comprueba el bit E y determina si puede dividir cuando el bit E vale 1 (se meterá un 1 en QN, siendo este el cociente) o no puede dividir cuando E vale 0 dicho número y requiere de hacer otra rotación, por tanto realizará la restauración del número para dejarlo igual a antes de realizar la suma en C2 para la comprobación.

3) A continuación, se expone el resultado con el código propuesto.

```
s2= A[0];
s3= A[0];
i=1;

for(i=1;i<n;i++){
    if(A[i]>s2){
        s2=A[i];
```

Resuelto por TeamGetppid()

```
        addi t0, zero, 1        // t0=i=1
        lw   s2, 0(s0)          // s2= A[0]
        lw   s3, 0(s0)          // s3= A[0]
loop:
        sll  t1, t0, 2          // t1= 4i
        add  t1, t1, s0         // t1= 4i + dirA
        lw   t1, 0(t1)          // t1= A[i]
        sgt  t2, t1, s2         // meto un 1 en t2 si A[i]>s2
        beq  t2, zero, else     // si t2=0 (no se cumple) voy a else.
        lw   s2, 0(t1)          // s2= A[i] // puede usarse tambien move s2,t1
        j    Fin                // salto a Fin
else:   slt  t3, t1, s3         // meto on 1 en t3 si A[i] < s3
        beq  t3, zero, Fin      // si t3= zero -> Fin
        lw   s3, 0(t1)          // s3= A[i] // move s3, t1
        j    Fin                // salto a Fin
Fin:
        addi t0, t0, 1          // i++
        slt  t4, t0, s1
        bne  t4, zero, loop
```

NOTA: Es posible realizarlo con funciones, obviamente el código cambia. Como recomendación es mejor primero realizarlo en un lenguaje como C y posteriormente pasarlo a MIPS.

El paso de lw s2, 0(t1) --- lw s3, 0(t1), es posible realizarlo también con move s2, t1. Esto es debido a que el valor de A[i] ya se encuentra en un registro como es t1 y move acepta copiar registros.

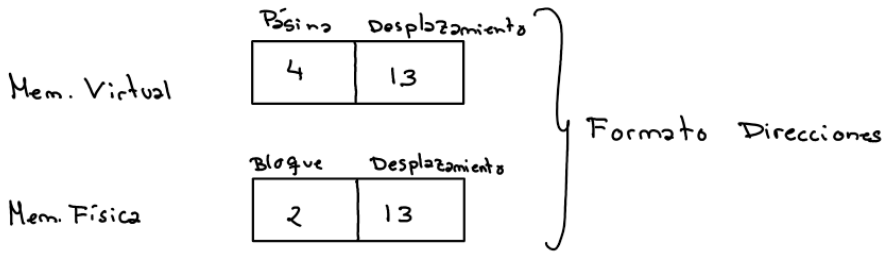
4) A continuación, se exponen los apartados:

APARTADO A

Espacio Direcciones: 128 kb → $2^7 \cdot 2^{10} = 2^{17}$

Espacio Memoña: 32 kb = $2^5 \cdot 2^{10} = 2^{15}$

Tamaño Página: 8 kb → $2^3 \cdot 2^{10} = 2^{13}$ → Siempre Tam. Página = Desplazamiento



Algoritmo LRU → 13, 8, 9, 8, 4, 13, 2, 8, 9

B0: 13
B1: 8
B2: ~~9~~ 2
B3: ~~4~~ 9

Página	Bloque
2	B2
8	B1
9	B3
13	B0

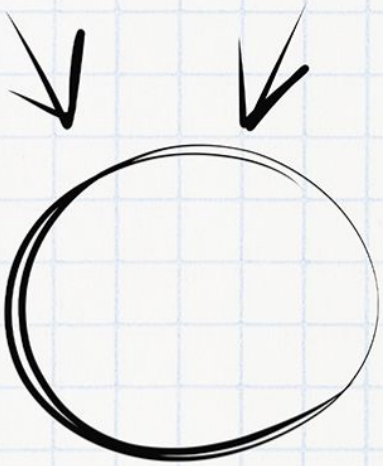
APARTADO B

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	0,99 € / mes	3,99 € / mes	7,99 € / mes

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

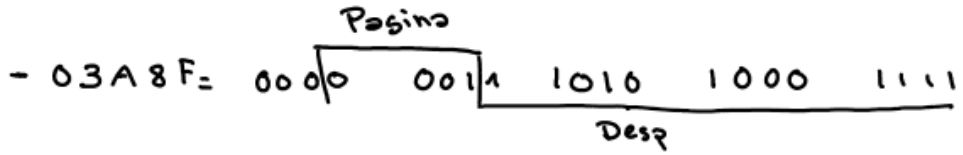


WUOLAH

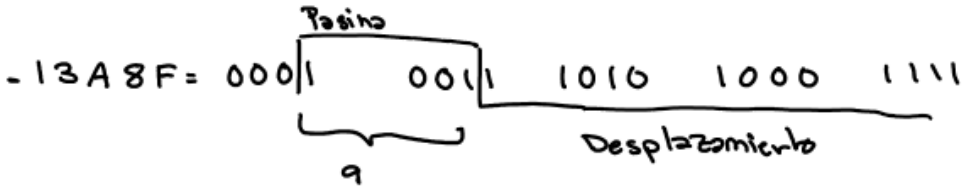
Los Rangos se obtienen de aquellas paginas que no están en memoria
Tenemos $2^4 = 16$ páginas - 4 (en memoria) = 12 páginas

	Pagina	Rango
P.0	0000	000 000 000 000 0
P.1	0001	111 111 111 111 1
P.3	0011	000 000 000 0000
P.7	0111	111 111 111 1111
P.10	1010	000 000 000 000 0
P.12	1100	111 111 111 111 1
P.14	1110	000 000 000 0000
P.15	1111	111 111 111 1111

APARTADO C



Fallo de Página, la pag. 1 no esta en memoria



5) Para resolver el ejercicio se ha planteado mediante la siguiente tabla LCB:

Resuelto por TeamGetppid()

S_2	S_1	S_0	Q_n	Q_{n+1}	Q_n	Z_{sc}	I	B	R	E
0	0	0					1	0	0	1
0	0	1					0	1	0	1
1	1	1					0	0	1	1
0	1	0	0	0	x	x	0	1	0	1
0	1	0	0	1	x	x	1	0	0	1
0	1	0	1	0	x	x	1	0	0	1
0	1	0	1	1	x	x	0	1	0	1
0	1	1	x	x	0	x	1	0	0	1
0	1	1	x	x	1	x	0	1	0	1
1	0	0	x	x	x	0	0	1	0	1
1	0	0	x	x	x	1	1	0	0	1

Con QnQn+1, controlamos que haga un bifurca cuando sea 00/11, pero como no podemos controlar que cuando hagamos incrementa vaya hacia izquierda o derecha, hemos de usar un bit adicional, pudiendo escoger entre Qn o Qn+1 (da igual), de modo que cuando uno de ellos sea 0 (haremos Bifurca o Incrementa) y el otro 1 (haremos lo contrario).

Control microprogramado, optimizado:

Dirección	Nº operación	Bit 2cb	Dir. Salto
Fetch+0	PC → Mar	000	—
+1	PC+1 → PC, M → GPR	000	—
+2	GPR(AD) → Mar, GPR(OP) → OPR	111	—
MLTB+0	M → GPR	000	—
+1	0 → Ac, 0 → Qn+1, n → Sc	000	—
+2	¿Qn Qn+1? Sc-1 → Sc	010	¿+6?
+3	¿Qn?	011	¿+5?
+4	Ac+BR → Ac	001	+6
+5	Ac+$\overline{BR}$ → Ac	000	——
+6	Ashr AQ	100	¿+2?
+7	—	001	Fetch+0

Control Micro-Cableado Optimizado:

Condición	Nº operación	Siguiente
t_0	PC → Mar	SR+1 → SR
t_1	PC+1 → PC, M → GPR	SR+1 → SR
t_2	GPR(AD) → Mar, GPR(OP) → OPR	SR+1 → SR
i0.t3	M → GPR	SR+1 → SR
i0.t4	0 → AC, 0 → Qn+1, n → SC	SR+1 → SR
i0.t5	SC-1 → SC	SR+1 → SR
i0.t5. $\overline{Qn}$. Qn+1	AC ← AC+BR	—
i0. t5. Qn. $\overline{Qn+1}$	AC ← AC+$\overline{BR}$+1	—
i0. t6	ashr A, QR	—
i0.t6 $\overline{Zsc}$	—	S → SR
i0.t6. Zsc	—	0 → SR

Expresiones de Control:

Micro-Operación	Expresión de Control
PC → Mar	T0
PC+1→PC	T1
M → GPR	T1 + i0.T3
GPR(AD)→MAR	T2
GPR(OP)→OPR	T2
0 → ACC	I0.t4
0→ Qn+1	I0.t4
N→SC	I0.t4
SC-1→SC	I0.t5
ACC+BR→ACC	I0.t5.$\overline{Qn}$.Qn+1
AC+$\overline{BR}$+1→AC	I0.t5.Qn.$\overline{Qn+1}$
ASHR A & QR	I0.t6
Load SR	I0.t6.ZsC + i0.t6.$\overline{ZSC}$ == **i0.t6**
SR +1 → SR	t0+t1+t2+t3+t4+t5