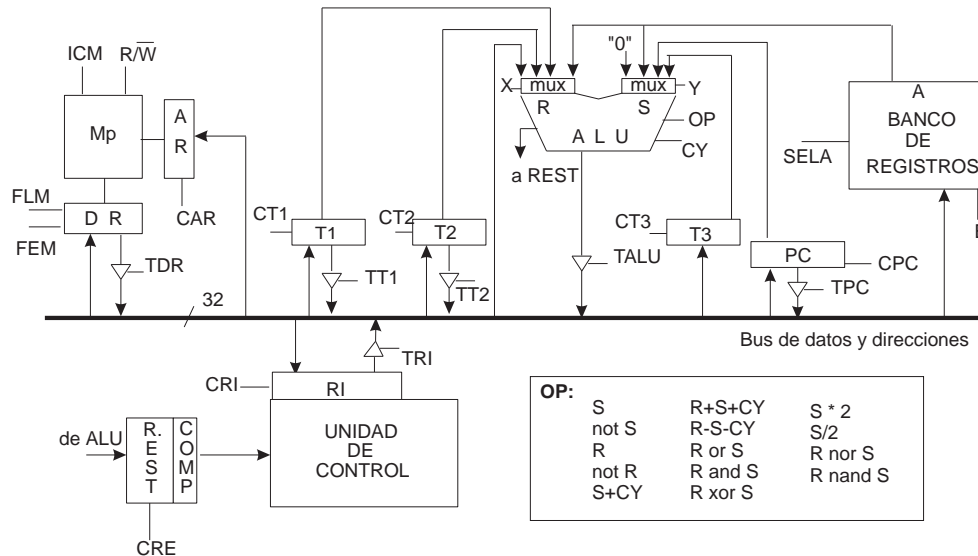


1 (4 puntos) La estructura del computador de 64 bits que aparece en la figura, con Unidad de control cableada y 15 registros de propósito general, utiliza el registro de propósito general R10 como puntero de pila, apuntando a la última posición ocupada de la cima de la pila. La pila crece hacia direcciones de memoria decrecientes y en ella se almacena la dirección de retorno de subrutina. Los accesos a memoria tienen una duración de dos ciclos de reloj y el direccionamiento es a nivel de palabra. Todos los incrementos o decrementos de registros se realizan a través de la ALU.



a) Expresa a nivel RT (transferencia entre registros) las operaciones elementales que se realizan durante la fase de fetch.

b) Para cada una de las siguientes instrucciones, exprese a nivel RT las operaciones elementales que se realizan, en cada ciclo de reloj, durante sus respectivas fases de ejecución.

CALL [.R3]

RET

BR \$16

c) Considerando los siguientes retardos en los dispositivos del computador:

- Multiplexor: 0,2 ut (unidad de tiempo)
- Puerta triestado: 0,1 ut
- Lectura o escritura de registro: 1 ut
- Lectura o escritura del banco de registros: 2 ut
- Operación de la ALU: 10 ut

Determine el periodo mínimo de reloj en este computador y calcule el tiempo que tardaría en ejecutarse cada una de las instrucciones del apartado anterior.

SOLUCIÓN

a) En el fetch se realizan las siguientes operaciones elementales:

$$f1: AR \leftarrow PC$$

f2: Accesso a memoria, $PC \leftarrow PC+1$

$$f3:DR \leftarrow M(AR)$$
$$f_4: \mathbb{R} \leftarrow \mathbb{R}$$

b) Las operaciones elementales realizadas, en la fase de ejecución, para cada instrucción son las siguientes:

CALL [.R3]

$$e1:AR, R10 \leftarrow R10 - 1$$

e2:DR \leftarrow PC

e3:Acceso a memoria, PC \leftarrow R3

e4:M(AR) \leftarrow DR

RET

e1:AR \leftarrow R10

e2:Acceso a memoria, R10 \leftarrow R10 + 1

e3:DR \leftarrow M(AR)

e4:PC \leftarrow DR

BR \$16

e1:T1 \leftarrow IR(desplazamiento)

e2:PC \leftarrow T1 + PC

c) El tiempo de ejecución se obtiene deduciendo, en primer lugar, el camino crítico que determinará el mínimo periodo de reloj. :

$$Caminocritico = Tck = t_{lectura}BR + tMx + tALU + ttriestado + t_{escritura}BR = (2+0, 2+10+0, 1+2)ut = 14,3ut$$

Por lo tanto, considerando un ciclo para la decodificación y, teniendo en cuenta, que el tiempo de ejecución es el producto del número de ciclos por el tiempo de ciclo, para cada instrucción se tiene:

$$t_{ejec}CALL = 9 \text{ ciclos} \cdot 14,3 \text{ ut/ciclo} = 128,7 \text{ ut}$$

$$t_{ejec}RET = 9 \text{ ciclos} \cdot 14,3 \text{ ut/ciclo} = 128,7 \text{ ut}$$

$$t_{ejec}BR = 7 \text{ ciclos} \cdot 14,3 \text{ ut/ciclo} = 100,1 \text{ ut}$$

2 (2 puntos) Un computador dispone de una memoria cache cuyo tiempo de acceso es de 2 ns y de una memoria principal en la que se emplean 60 ns para leer o escribir una palabra (32 bits). El tamaño del bloque es de cuatro palabras. Determine justificadamente el tiempo de escritura cuando hay un fallo de memoria cache suponiendo que:

a) La política de escritura es CBWA (aplazada con actualización) y

a.1) El bloque a reemplazar está modificado.

a.2) El bloque a reemplazar no está modificado

b) La política de escritura es WTWNA (inmediata sin actualización).

SOLUCIÓN

a) Con política de escritura CBWA el tiempo de escritura cuando se produce fallo de cache y el bloque está modificado incluye un primer acceso a la cache, la transferencia del bloque modificado a la memoria principal, la transferencia del nuevo bloque de la memoria principal a la memoria cache y un último acceso a la memoria cache para escribir la palabra. Así:

$$t_{acceso} = (t_{Mca} + t_{BlqMca \rightarrow Mp} + t_{BlqMp \rightarrow Mca} + t_{Mca}) = 2 + 240 + 240 + 2 = 484 \text{ ns}$$

Cuando el bloque a reemplazar no está modificado no se realiza la transferencia de dicho bloque a la memoria principal. Así:

$$t_{acceso} = (t_{Mca} + t_{BlqMp \rightarrow Mca} + t_{Mca}) = 2 + 240 + 2 = 244 \text{ ns}$$

b) Con política de escritura WTWNA se escribe simultáneamente en cache y memoria principal. Así:

$$t_{\text{acceso}} = t_{Mp} = 60 \text{ ns}$$

3 (4 puntos) Programe en ensamblador del 88110 la función **reemplazar** que recibe tres parámetros en la pila:

- **m**: Es una matriz cuadrada de orden **n** (**n** filas y **n** columnas) de enteros. La matriz está almacenada por filas. Se pasa por dirección.
- **v**: Es un vector de **n** elementos que contiene los valores con los que se reemplaza la matriz. Se pasa por dirección.
- **n**: Es un entero que contiene el orden de la matriz. Se pasa por valor.

Esta función recorrerá la matriz y, cada elemento nulo ubicado en una columna (**i**), lo reemplazará con el elemento **i** del vector **v** y devolverá en **r29** el número de elementos que ha reemplazado.

El resultado de invocar a la subrutina **reemplazar** con la matriz que aparece a la izquierda con el parámetro **v** conteniendo el valor (10,9,8,7) es el que se indica a la derecha. El registro **r29** contendrá el valor 7 como resultado de la llamada, puesto que ha modificado los 7 elementos señalados en **negrita**.

$$m = \begin{pmatrix} 10 & 0 & 1 & 5 \\ -10 & 2 & 1 & 3 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & 4 & 0 \end{pmatrix}$$

$$m = \begin{pmatrix} 10 & \mathbf{9} & 1 & 5 \\ -10 & 2 & 1 & 3 \\ \mathbf{10} & -2 & \mathbf{8} & \mathbf{7} \\ \mathbf{10} & \mathbf{9} & 4 & \mathbf{7} \end{pmatrix}$$

SOLUCIÓN

La subrutina **reemplazar** que se muestra a continuación realiza un recorrido por la matriz mediante dos bucles anidados: uno para las filas de la matriz (externo) y otro para las columnas. Para cada elemento, se comprueba si es 0 y, en ese caso, se reemplaza por el valor del elemento correspondiente de **v** y se incrementa el registro **r29**. Cuando se ha alcanzado el final de una fila, se decrementa el número de filas, se carga el orden de la matriz en el número de columnas y se vuelve el puntero al vector al comienzo del mismo.

```
reemplazar:  ld r4,r30,8      ; El Orden de la matriz se copia en r2 y r3.
             or r3, r4, r4   ; r2 y r3 llevan los contadores de número de columnas y
             or r29,r0,r0    ; filas respectivamente y r29 el número de elementos modificados
             ld r20,r30,0    ; Dirección de comienzo de Matriz
             ld r21,r30,4    ; Dirección de comienzo de Vector
bucle:      or r2, r4, r4    ; Inicializa el número de columnas
             or r22,r21,r21  ; Cada iteración del bucle externo se pone el puntero al
                               ; vector al comienzo del mismo
bucle_i:    ld r5,r20,r0     ; Carga el elemento de la matriz
             cmp r7,r5,r0
             bbl ne,r7,cont
             ld r7,r22,r0    ; El elemento es nulo, por lo que se carga el elemento del vector
             st r7,r20,r0    ; y lo reemplaza en la matriz
             addu r29,r29,1   ; Incrementa el número de elementos cambiados
cont:       addu r20,r20,4    ; Incrementa puntero de matriz
             addu r22,r22,4   ; Incrementa puntero de vector
             subu r2,r2,1     ; Decrementa número columnas
             cmp r7,r2,r0
             bbl ne,r7,bucle_i
             subu r3,r3,1     ; Si el número de columnas es nulo, se decrementa número filas
             cmp r7,r3,r0     ; Si el número de filas es nulo, se acaba
             bbl ne,r7,bucle
             jmp(r1)
```