

Apellidos \_\_\_\_\_

Nombre \_\_\_\_\_

## Normas del examen

- La duración del examen es de 2 horas.
- Resuelva el examen en las mismas hojas de los enunciados.
- **Se penalizará la ausencia de comentarios.**
- **No puedes utilizar ni calculadora ni documentación.**

## Estudiantes con la asignatura F06 ya aprobada

- Marca la casilla si deseas convalidar la asignatura F06 por el primer parcial de ésta.  
Pasar por el despacho de Rafael Mayo en las dos próximas semanas en su horario de tutorías para confirmar la solicitud.

## Cuestiones

- 1,25 1. ¿Cómo se establece la prioridad entre los diferentes dispositivos en un sistema de interrupciones no vectorizadas?
- En este tipo de sistema de gestión de interrupciones la prioridad se establece según el código de la rutina de servicio de interrupción. Dicha rutina de servicio de interrupción debe leer el registro de estado de los controladores para averiguar cual de ellos es el que ha generado la interrupción. Así pues dicho código es el que marca la prioridad.*

- 1,25 2. ¿Qué condiciones hacen que un sistema de memoria cache con organización directa sea más lento que un sistema sin memoria cache?

*En una memoria cache con organización directa, varios bloques de memoria central tienen asignada la misma línea de memoria cache donde pueden copiarse. Si un programa tiene un patrón de acceso a memoria tal que se direccionan consecutivamente distintos bloques de memoria que tiene asignada la misma línea de cache se estarán produciendo continuos fallos de cache lo que hará que el sistema pueda llegar a ser más lento que si no se tuviese dicha cache*

- 1,25 3. ¿En qué consiste la jerarquía de memoria?. ¿Sobre qué principios se basa?. Indica los diferentes niveles dentro de dicha jerarquía.

*La jerarquía de memoria consiste en la construcción del sistema de memoria a partir de distintos elementos de memoria que se disponen dependiendo de su velocidad y capacidad, utilizando los de menor capacidad y mayor velocidad para los datos que están siendo utilizados en cada momento y los de mayor capacidad y menor velocidad para los datos que en ese momento no están siendo utilizados.*

*El principio en que se basa esta organización de la memoria es el de localidad de las referencias, en sus dos vertientes, temporal y espacial. La vertiente temporal indica que la probabilidad de referenciar una determinada posición de memoria aumenta cuanto menos tiempo hace que ha sido referenciada. Por otro lado la vertiente espacial indica que la probabilidad de referenciar una determinada posición de memoria aumenta cuanto más próxima está a la última posición referenciada.*

*Los niveles básicos que se tienen en la organización jerárquica de la memoria son: los registros, la memoria cache, la memoria central y los dispositivos de almacenamiento secundario.*

Apellidos \_\_\_\_\_

Nombre \_\_\_\_\_

1,25

4. ¿En qué consiste la segmentación de los procesadores?. ¿Porqué los procesadores segmentados son más rápidos que los no segmentados si utilizan la misma tecnología de fabricación? Indica brevemente los problemas que se plantean cuando se aplica la segmentación.

*La segmentación consiste en dividir los elementos de la ruta de datos según las fases de ejecución de las instrucciones, generando una ruta de datos compuesta de una secuencia de unidades funcionales especializadas unidas mediante unos registros. Cada una de esas unidades genera en cada ciclo de reloj un resultado parcial que se toma como entrada, en el siguiente ciclo, de la siguiente unidad en la ruta de datos.*

*Los procesadores diseñados de esta forma son más rápidos que los no segmentados porque una vez una unidad de la ruta de datos ya ha realizado el trabajo para la instrucción  $i$ , en el siguiente ciclo ya se puede utilizar para realizar el trabajo para la instrucción  $i+1$  por lo que en un mismo ciclo de reloj pueden estar realizándose trabajo para tantas instrucciones como segmentos se hayan realizado en la ruta de datos. Así pues, en el mejor de los casos en cada ciclo de instrucción se finaliza una instrucción. Por ejemplo, si todas las instrucciones de una determinada ISA tuviesen un CPI de 4, la ejecución de 5 instrucciones en el procesador no segmentado costaría  $5*4=20$  ciclos de reloj, mientras que si se implementa el procesador segmentado con 4 segmentos, esas 5 instrucciones se ejecutarían en 8 ciclos de reloj  $4 + (5-1)$  por lo que en un mismo ciclo de reloj pueden estar realizándose trabajo para la tantas instrucciones como segmentos se hayan realizado en la ruta de datos.*

*Los problemas que plantean este tipo de implementaciones del procesador son:*

- *Problemas en la gestión de saltos, sobre todo en los condicionales.*
- *Problemas de dependencias de datos, que una instrucción vaya a utilizar un dato que todavía no ha sido generado por una instrucción anterior.*
- *Problemas de compartición de elementos de la arquitectura, por ejemplo que en un mismo ciclo la ejecución de dos instrucciones distintas requiera el acceso a memoria central.*

**El contenido de esta página no será corregido.**

*Puedes utilizarla para las anotaciones que necesites realizar.*

Apellidos \_\_\_\_\_  
 Nombre \_\_\_\_\_

- 3 5. Desarrolla una rutina en ensamblador del R2000 que codifique la siguiente rutina en C, realizando el paso de parámetros según el convenio establecido.

```
int iguales ( int *A, /* vector A de dimension n */
             int *B, /* vector B de dimension n */
             int n) /* dimension n de los vectores */
{
    int i,cont;

    cont=0;
    for (i=0;i++;i<n)
        if (A[i]==B[i]) cont++;
    return(cont);
}
```

Segun el criterio establecido los parámetros se pasarán de la siguiente forma:

- El parámetro A en el registro \$a0 y por referencia, es decir en el registro \$a0 se tendrá la dirección de inicio del vector A.
- El parámetro B en el registro \$a1 y por referencia, es decir en el registro \$a1 se tendrá la dirección de inicio del vector B.
- El parámetro n en el registro \$a2 y por valor, es decir en el registro \$a2 se tendrá el valor n.

```
iguales:  addi $sp, $sp, -16      ; Se apilan los registros
          sw   $t0, 12($sp)    ; que se van a modificar.
          sw   $t1, 8($sp)     ;
          sw   $t2, 4($sp)     ;
          sw   $t3, 0($sp)     ;

          and  $t0, $t0, $zero  ; t0=0 (cont, se implementa sobre t0)
          and  $t1, $t1, $zero  ; i=0 (i, se implementa sobre t1)

para:     beq  $t1, $a2, finpara ; el for finaliza cuando i=n, es decir
          ; t1 = a2

          sll  $t2, $t1, 2      ; t2 = t1 * 4, ya que los elementos
          ; de los vectores ocupan 4 bytes
          add  $t2, $t2, $a0    ; t2 = direccion de memoria del
          ; elemento A[i]
          lw   $t2, 0($t2)     ; t2 = A[i]

          sll  $t3, $t1, 2      ; t3 = t1 * 4, ya que los elementos
          ; de los vectores ocupan 4 bytes
```

## Estructura y Tecnología de Computadores (IG09) 2º Parcial 12 de Septiembre de 2002

```
add $t3, $t3, $a1      ; t3 = direccion de memoria del
                       ; elemento B[i]
lw  $t3, 0($t3)        ; t3 = B[i]

bne $t3, $t2, finssi   ; solo se incrementa si son iguales.
addi $t0, $t0, 1       ;

finssi:  addi $t1, $t1, 1

        j para          ; vuelve al inicio del bucle

finpara: or  $v0, $t0, $zero ; el resultado se devuelve en v0

lw  $t0, 12($sp)       ; Se rescata el valor de los
lw  $t1, 8($sp)        ; registros modificados
lw  $t2, 4($sp)        ;
lw  $t3, 0($sp)        ;
addi $sp, $sp, 16      ;

jr  $ra                ; retorno de subrutina
```

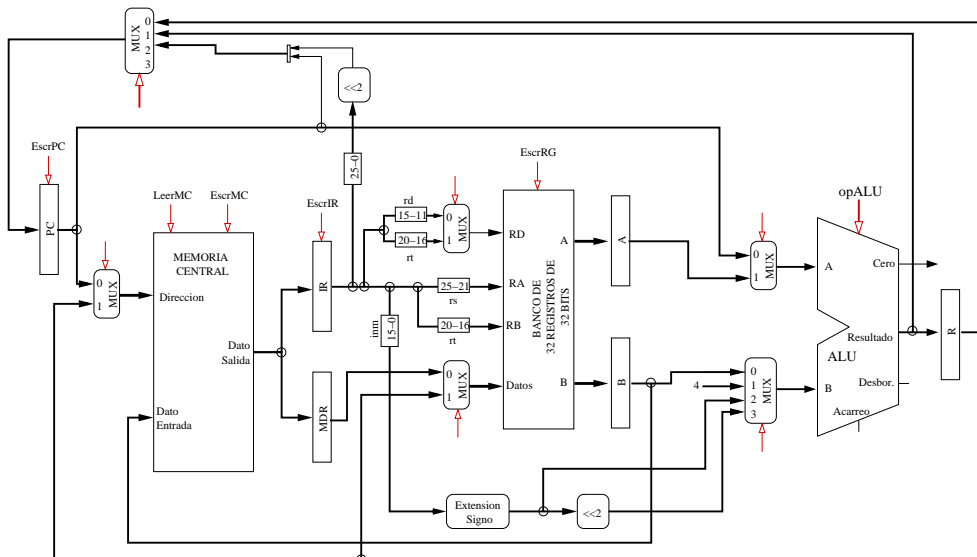
Apellidos \_\_\_\_\_  
 Nombre \_\_\_\_\_

- 2 6. Estudia la posibilidad de modificar la ruta de datos estudiada, para implementar como instrucción la pseudoinstrucción

`move reg1, reg2`

que copia el contenido del registro `reg2` en el registro `reg1`. Se debe realizar el menor número de modificaciones posible. clase.

Nota: Téngase en cuenta que lo que primero se debe hacer es optar por un determinado formato para dicha instrucción (R,I o J).



Hay dos posibilidades a la hora de afrontar este problema

- La primera consiste en asimilar esta pseudo instrucción con alguna de las que ya existen, por ejemplo:
  - `add $t0, $t1, $0`
  - `addi $t0, $t1, 0`
  - `or $t0, $t1, $0`
  - etc.

con lo que el problema ya está resuelto, simplemente habría que indicar que en la codificación un determinado registro es el \$0 o el dato inmediato es 0

- La segunda, que consigue una ejecución más rápida, consiste en realizar una ligera modificación a la ruta de datos, simplemente hacer que la salida del registro B esté conectada con el multiplexor que da acceso a la entrada de datos del banco de registros. De esta forma en

**Estructura y Tecnología de Computadores (IG09) 2º Parcial** 12 de Septiembre de 2002

3 ciclos de reloj se puede realizar el movimiento de datos entre registros en lugar de en los cuatro que se necesitan si se opta por la primera alternativa.

Cualquiera de las dos alternativas se considera igual de válida en la corrección.