Eres Estudiante

Exámenes

Aviso: Recomendamos **no abrir el examen** en más de una pestaña del navegador, se puede producir pérdida de datos en el envío.

Importante: Los exámenes no aparecen automáticamente a la hora señalada. Si accede antes de la hora, **debe recargar la herramienta** a la hora de inicio del examen.

Cuestionario AEC-P2yP3 - SubGrupo 1.2

Volver a la Lista de Exámenes

Parte 1 de 3 **3.5 / 3.5 Puntos**

Crea un fichero de texto llamado **cuestionario_p2y3_a.s** y copia el código que se proporciona.

Ejecutar el programa ciclo a ciclo hasta finalizar la primera iteración del bucle para el procesador **DLX sin adelantamiento**, considerando la inserción de ciclos de parada para los riesgos de datos y la estrategia de **predicción *predict-not-taken*** para los riesgos de control.

```
.data
    x: .word 0,1,2,3,4,5,6,7,8,9
       .word 10,11,12,13,14,15
    a: .word 0,1,2,3,4,5,6,7,8,9
       .word 10,11,12,13,14,15
    ; Vector y
    ; 16 elementos son 64 bytes.
    y: .space 64
; El código
.text
start:
    add r4,r0,#64 ; 16*4
    add r2,r0,y
    add r1,r0,x
    add r5,r0,a
loop:
    lw r6, 0(r5)
    lw r3, 0(r1)
    add r6, r3, r6
    add r5, r5, #4
    sub r4, r4, #4
    add r7, r6, #8
    add r1, r1, #4
    subi r7, r6, #8
    sw 0(r2), r6
    add r2, r2, #4
    add r3, r3, #1
    add r7, r7, r2
    bnez r4,loop
    trap #0 ; Fin de programa
```

Preguntas 1 de 18	0.25

0.25 Puntos

Especifica la **1ª instrucción**, en orden de programa, que detiene el pipeline por un riesgo de datos: ✔lw r6,0(r5)
Número de ciclos de parada: ✔2

Respuesta correcta: lw r6, 0(r5), 2

Preguntas 2 de 18	0.25

0.25 Puntos

Especifica la **2ª instrucción**, en orden de programa, que detiene el pipeline por un riesgo de datos: ✔add r6,r3,r6
Número de ciclos de parada: ✔2

Respuesta correcta: add r6, r3, r6, 2

Preguntas 3 de 18	0.25

0.25 Puntos

Especifica la **3ª instrucción**, en orden de programa, que detiene el pipeline por un riesgo de datos: ✔add r7,r7,r2
Número de ciclos de parada: ✔1

Ejemplo de respuesta correcta:
Instrucción: add r2, r0, y
Número de ciclos de parada: 6

Respuesta correcta: add r7, r7, r2, 1

Preguntas 4 de 18	0.25

0.25 Puntos

Especifica la instrucción que genera un riesgo de control: ✔bnez r4,loop
Número de ciclos de parada/vaciado pipeline: ✔3

Respuesta correcta: bnez r4, loop, 3

Preguntas 5 de 18	0.5

0.5 Puntos

Para la primera iteración del bucle (e ignorando todas las instrucciones anteriores al bucle) específica el número de instrucciones ejecutadas: ✔13 y el número de ciclos de parada (**OJO**: no contar los ciclos de parada insertados por instrucciones anteriores al bucle): ✔8 .

Respuesta correcta: 13, 8

Preguntas 6 de 18	0.5

0.5 Puntos

Calcula **manualmente** el número **TOTAL** de instrucciones ejecutadas por el programa (considerando **todas** las instrucciones del código).
No basta con dar el número final, sino la expresión que te ha llevado a dicho número.
Ejemplo de respuesta correcta (antes del bucle + dentro del bucle + después del bucle): 8 + 20 * 10 + 3 = 211

Total NI: ✔4+13*16+1=213

Respuesta correcta: 4 + 13 * 16 + 1 = 213 | 4 + (13 * 16) + 1 = 213 |

Preguntas 7 de 18	0.5

0.5 Puntos

Calcula **manualmente** el número **TOTAL** de ciclos de parada en el programa (considerando **todas** las instrucciones del código).
No basta con dar el número final, sino la expresión que te ha llevado a dicho número.
Ejemplo de respuesta correcta (antes del bucle + dentro del bucle + después del bucle): 4 + 20 * 16 + 3 = 327

Total ciclos de parada: ✔8+6*14+3=95

Respuesta correcta: 0 + 8 + 6*14 + 3 + 0 = 95 | 8 + 6*14 + 3 = 95 | 0 + 8 + (6*14) + 3 + 0 = 95 | 8 + (6*14) + 3 = 95

Preguntas 8 de 18	1.0

1.0 Puntos

¿Si activamos **forwarding/adelantamientos** logramos eliminar todos los ciclos de parada por riesgos de datos? Si (1), No (2). ✔2

Ahora, con forwarding/adelantamiento activo, el número de ciclos de parada en la primera iteración del bucle sería de ___ ciclos. ✔4

Respuesta correcta: 2, 4

Parte 2 de 3 - Predicción dinámica de saltos **2.5 / 3.0 Puntos**

Preguntas 9 de 18	2.0

2.5 Puntos

Pulse para ver instrucciones adicionales

Diseñar y compilar un predictor de saltos que use **un único** contador saturado global de 3 bits para todos los saltos. Dicho contador se inicializa a 4. (Usa siempre **cuatro decimales** en las respuestas, tal y como aparece en la salida del simulador.)

¿Qué tasa de acierto en la predicción se obtiene para lbm? ✔90.0055 % ¿Y para mcf? ✔59.7097 %

Diseñar y compilar un predictor de saltos basado en contadores saturados de 4 bits (inicializados a 8) y con un total de 16 contadores. Para saber qué contador usa cada salto se define una función *hash* que dada la dirección del salto nos diga cuál de los 16 contadores debe usar de la siguiente forma: PC % 16. (Usa siempre **cuatro decimales** en las respuestas, tal y como aparece en la salida del simulador.)

¿Qué tasa de predicción se obtiene para `lbm`? ✔89.9918 % ¿Y para `mcf`? ✔99.4697 %

Calcula la mejora relativa de ambas aplicaciones al aumentar el tamaño de la tabla a 32 entradas, ¿que aplicación se beneficia más en % al aumentar el tamaño de la tabla, lbm(1) o mcf(2)? ✖2

Respuesta correcta: 90.0055, 59.7097, 89.9918, 99.4697, 1

Preguntas 10 de 18	0.0

0.0 Puntos

Copie aquí el código fuente de los predictores implementados anteriormente.

```
//PREDICTOR 1
#include "ooo_cpu.h"
#define TAKEN true
#define NOT_TAKEN false
int cont;
void O3_CPU::initialize_branch_predictor()
{
cont=4;
}
uint8_t O3_CPU::predict_branch(uint64_t pc)
{
if (cont>=4){
return TAKEN;
}
else if (cont<4){
return NOT_TAKEN;
}
}
void O3_CPU::last_branch_result(uint64_t pc, uint8_t taken)
{
if (taken && cont<7){
cont++;
}
else if (!taken && cont>0){

cont--;
}
}
//PREDICTOR 2
#include "ooo_cpu.h"
#define TAKEN true
#define NOT_TAKEN false
int cont[16];
void O3_CPU::initialize_branch_predictor()
{
for (int i=0;i<16;i++){
cont[i]=8;
}
}
uint8_t O3_CPU::predict_branch(uint64_t pc)
{
int hash=pc%16;
if (cont[hash]>=8){
return TAKEN;
}
else if (cont[hash]<8){
return NOT_TAKEN;
}
}
void O3_CPU::last_branch_result(uint64_t pc, uint8_t taken)
{
```

```
int hash=pc%16;
if (taken && cont[hash]<15){
cont[hash]++;}
else if (!taken && cont[hash]>0){
cont[hash]--;
}
}
```

Preguntas 11 de 18	0.5

0.5 Puntos

Usando un predictor con contadores saturados, decimos que una aplicación sufre de *aliasing*:

- ✔ **A.** **cuando al aumentar el tamaño de los contadores observamos una mejora en la precisión del predictor.**
- ✔ **B.** **cuando al aumentar el número de entradas de la tabla de predicción observamos que no mejora la precisión del predictor.**
- ✔ **C.** **cuando al aumentar el número de entradas de la tabla de predicción observamos una mejora en la precisión del predictor.**
- ✔ **D.** **cuando al aumentar el tamaño de los contadores observamos que no mejora la precisión del predictor.**

Respuesta correcta: C

Parte 3 de 3 - Segmentación de Punto Flotante **3.5 / 3.5 Puntos**

Abre el simulador Simula3MS y copia el siguiente código en la ventana de código, guárdalo en un fichero de texto llamado **cuestionario_p2y3_b.s**:

```
.data
a:  .float 1
b:  .float 2
va: .double 1,2,3,4,5,6,7,8,9,10
vb: .double 11,12,13,14,15,16,17,18,19,20
z: .space 80
.text
.globl main
main:
      la $t1, va            # carga la dir vector va en $t1
      la $t2, vb            # carga la dir vector vb en $t2
      la $t7, vb            # carga la dir vector vb en $t7 para comprobar el final del bucle
      la $t5, z             # carga la dir z en $t5
      la $t6, a             # carga la dir de "a" en $t6
      lwc1 $f1, 0($t6)      # carga "a" en f0
      cvt.d.s $f0, $f1      # convertimos "a" en DP y lo guardamos en $f0:$f1
      la $t6, b             # carga la dir de "b" en $t6
      lwc1 $f3, 0($t6)      # carga "b" en f3
      cvt.d.s $f2, $f3      # convertimos "b" en DP y lo guardamos en $f2:$f3
loop:
      lwc1 $f8, 0($t1)
      lwc1 $f9, 4($t1)      # f8:f9=va[i]
      lwc1 $f11,4($t2)      # f10:f11=vb[i]
      lwc1 $f10,0($t2)
      div.d $f10,$f10,$f2   # f10=vb[i]*b
      mul.d $f8,$f8,$f0     # f8=va[i]*a
      addi $t1,$t1,8        # actualizamos todos los indices
      addi $t2,$t2,8
      addi $t5,$t5,8
      add.d $f16,$f8,$f10   # f16 = va[i]*a + vb[i]*b
      swc1 $f16, -8($t5)    # almacenamos la parte baja del resultado z[i]
      swc1 $f17, -4($t5)    # almacenamos la parte alta del resultado z[i]
      bne $t1, $t7, loop    # comprobamos si hemos terminado el bucle
      addi $v0, $0, 10
      syscall               # llamada para salir del programa
```

Preguntas 12 de 18	0.25

0.25 Puntos

Usando el simulador *Simula3MSv5.01* ejecuta ciclo a ciclo hasta terminar la **primera iteración** del bucle. En la configuración selecciona procesador segmentado básico con adelantamiento, donde el salto se resuelve mediante la opción "Salto/Un Hueco/Predecir no tomado" con las siguientes unidades funcionales:

- 2 unidad sumadora (no segmentada) de punto flotante con 3 ciclos de latencia.
- 2 unidad multiplicadora (no segmentada) de punto flotante con 3 ciclos de latencia.
- 1 unidad divisora (no segmentada) de punto flotante con 8 ciclos de latencia.

Observa el avance de las instrucciones a lo largo del cauce, así como la inserción de ciclos de parada cuando se detectan riesgos. Para las siguientes cuestiones vamos a centrarnos en la 1ª iteración del cuerpo de bucle y nos olvidamos de las instrucciones de antes del bucle.

Indica la **1ª instrucción** del cuerpo del bucle que inserta ciclos de parada (**OJO**: En caso de riesgo estructural, indicar la instrucción que se bloquea esperando un recurso): ✔div.d $f10,$f10,$f2

Cuántos ciclos de parada inserta: ✔1

Tipo de riesgo (datos o estructural): ✔datos

Respuesta correcta: div.d $f10,$f10,$f2, 1, datos

Preguntas 13 de 18	0.25

0.25 Puntos

Indica la **2ª instrucción** del cuerpo del bucle que inserta ciclos de parada (**OJO**: En caso de riesgo estructural, indicar la instrucción que se bloquea esperando un recurso): ✔add.d $f16,$f8,$f10

Cuántos ciclos de parada inserta: ✔3

Tipo de riesgo (datos o estructural): ✔datos

Respuesta correcta: add.d $f16,$f8,$f10, 3, datos

Preguntas 14 de 18	0.25

0.25 Puntos

Indica la **3ª instrucción** del cuerpo del bucle que inserta ciclos de parada (**OJO**: En caso de riesgo estructural, indicar la instrucción que se bloquea esperando un recurso): ✔swc1 $f16, -8($t5)

Cuántos ciclos de parada inserta: ✔2

Tipo de riesgo (datos o estructural): ✔datos

Respuesta correcta: swc1 $f16, -8($t5), 2, datos

Preguntas 15 de 18	0.5

0.5 Puntos

Calcula **manualmente** el número **TOTAL** de instrucciones ejecutadas (ahora sí consideramos **todas** las instrucciones del código).
No basta con dar el número final, sino la expresión que te ha llevado a dicho número.
Ejemplo de respuesta correcta (antes del bucle + dentro del bucle + después del bucle): 4 + 23*5 + 3 = 122

Total NI = ✔16+(13*10)+1=147

Respuesta correcta: 16+13*10+2=148 | 16+13*10+1=147 | 16+(13*10)+2=148 | 16+(13*10)+1=147 | 10+13*10+2=142 | 10+13*10+1=141 | 10+(13*10)+2=142 | 10+(13*10)+1=141

Preguntas 16 de 18	0.5

0.5 Puntos

Calcula **manualmente** el número **TOTAL** de ciclos de parada (ahora sí consideramos **todas** las instrucciones del código).
No basta con dar el número final, sino la expresión que te ha llevado a dicho número.
Ejemplo de respuesta correcta (antes del bucle + dentro del bucle + después del bucle): 4 + 20*10 + 2 = 206

Total ciclos de parada = ✖2+(6*10)+9=71

Respuesta correcta: 1+7*10=71 | 1+(7*10)=71 | 1+7*10=71 | 1+(7*10)=71 | 2+7*9+6=71 | 2+(7*9)+6=71

Preguntas 17 de 18	0.5

0.5 Puntos

Calcula **manualmente** el número **TOTAL** de ciclos de ejecución (ahora sí consideramos **todas** las instrucciones del código).
No basta con dar el número final, sino la expresión que te ha llevado a dicho número.
Ejemplo de respuesta correcta (antes del bucle + dentro del bucle + después del bucle): 4 + 32*5 + 3 = 167

Total ciclos de ejecución = ✖147+71+5=223

Respuesta correcta: 17+22+20*9+4=223 | 17+22+20*9+3=222 | 17+22+(20*9)+4=223 | 17+22+(20*9)+3=222 | 147+71+4=222 | 148+71+4=223

Preguntas 18 de 18	1.25

1.25 Puntos

¿Existen riesgos estructurales con la configuración y código proporcionados? Si (1), No (2). ✔2

¿Cuantas unidades de suma con 3 ciclos de latencia debemos tener para minimizar los recursos y aún así no tener riesgos estructurales? ✔1 .

¿Cuantas unidades de multiplicación con 3 ciclos de latencia debemos tener para minimizar los recursos y aún así no tener riesgos estructurales? ✔1 .

Respuesta correcta: 2, 1, 1