

1. (5 puntos) Pretendemos resolver los riesgos de datos provocados por la siguiente secuencia de instrucciones:

```

1      .ireg 10,20,0,0,0
2      .data
3 a:    .dword 10
4      .text
5      ld r3 , a(r0)
6      dadd r4 , r3 , #5
7      dsub r5 , r3 , r4

```

En este ejercicio se pide completar el siguiente código del simulador del MIPS utilizado en prácticas para detectar y resolver únicamente dichos riesgos mediante cortocircuito y, si además fueran necesarios, ciclos de parada.

NOTA: Recuerda que están disponibles siguientes macros para analizar determinados campos de una instrucción:

- `es_load(inst)`. Indica que la instrucción `inst` es de carga.
- `lee_Rfte1(inst)`. Indica que la instrucción `inst` utiliza un registro fuente 1 válido.
- `lee_Rfte2(inst)`. Indica que la instrucción `inst` utiliza un registro fuente 2 válido.
- `escribe_Rdst(inst)`. Indica que la instrucción `inst` utiliza un registro destino válido.

Y que el simulador dispone de las siguientes señales de control:

- `IFstall, IDstall`: Mantiene la instrucción en la etapa correspondiente, entregando además una instrucción `nop` a la etapa siguiente.
- `IFnop, IDnop, EXnop`: Convierte la instrucción de la etapa correspondiente en `nop`.

```

1  /* Logica de control en la fase de decodificacion para realizar la deteccion
2     de riesgos y activar ciclos de parada */
3
4  void detectar_riesgos_datos(void) {
5      switch (solucion_riesgos_datos) {
6
7          case cortocircuito:
8              /* Riesgo entre LOAD en EX e ID */
9              if (_____ COMPLETAR CON CODIGO_1 _____) {
10                 IFstall = SI;
11                 IDstall = SI;
12             } /* endif */
13             break;
14             ...
15         }
16     }

```

```

1  /* Logica de control para activar cortocircuitos a traves del multiplexor
2     superior de la ALU */
3
4  dword mux_ALUsup(dword npc, dword ra, dword mem, dword wb) {
5
6      switch (solucion_riesgos_datos) {
7          case cortocircuito:
8              /* WBtoEX */
9              /* _____ INICIO CODIGO_2 _____ */

```

```

10         if (lee_Rfte1(ID.EX) && escribe_Rdst(MEMWB)&&
11             (ID.EX.IR.Rfuente1 == MEMWB.IR.Rdestino)) {
12             /* ----- FIN CODIGO_2 ----- */
13             WBaEXalu_s = SI;
14             result = wb;
15         } /* endif */
16         break;
17         ...
18     }
19 }

```

```

1  /* Logica de control para activar cortocircuitos a traves del multiplexor
2     inferior de la ALU */
3
4  dword mux_ALUinf(dword npc, dword ra, dword mem, dword wb) {
5
6  switch (solucion_riesgos_datos) {
7      case cortocircuito:
8          /* MEMtoEX */
9          if (----- COMPLETAR CON CODIGO_3 -----) {
10              MEMaEXalu_i = SI;
11              result = mem;
12          } /* endif */
13          break;
14          ...
15      }
16  }

```

En particular, se pide:

- a) **2 puntos** Escribir el CODIGO\_1 y explicar qué resuelve en la secuencia de instrucciones.
- b) **1 punto** Explicar qué resuelve el CODIGO\_2 en la secuencia de instrucciones.
- c) **2 puntos** Escribir el CODIGO\_3 y explicar qué resuelve en la secuencia de instrucciones.