# Proyecto 1 – 2023 – Gestión de riesgos y excepciones en MIPS

Pablo Ernesto Angusto Delgado

Miguel Aréjula Aísa

sábado, 25 de marzo de 2023

## Breve resumen

*(orientativamente, unas 200 palabras)*

## Verificación de los requisitos funcionales:

*Los apartados en los que se indica “Completar con otros casos representativos” o con ejemplos deben de incluir:*

* *Breve título indicando el objetivo del programa de prueba o testbench*
* *El código en ensamblador y el nombre de los ficheros RAM (datos y/o instrucciones) del test que se acompañan en la entrega, para el caso de que necesitemos comprobarlos.*
* *Comentario del test. Puede simplemente añadirse comentarios a las instrucciones en el código, incluir un cronograma de la ejecución, y/o una captura de pantalla del lugar o lugares del waveform de la simulación que muestren lo que ocurre.*

RF1 cambio a modo excepción

**Descripción:** Si el procesador está en modo usuario y recibe IRQ, ABORT o UNDEF pasa al modo correspondiente y ejecuta la rutina que indica la tabla de vectores de excepción. Si está en modo excepción ignora las señales hasta volver a modo usuario.

¿**Cómo**? Explicar brevemente la gestión del PC al aceptar la excepción.

**Verificación**:

* IRQ: banco de pruebas test\_IRQ en el que se realizan diversas IRQs. Todas ellas se atienden si y sólo si el procesador está en modo usuario. Se ejecuta una RTI que contabiliza el número de IRQs. Se verifica su funcionamiento correcto.
* Data Abort: en la memoria de instrucciones se incluyen dos bancos de pruebas en el que se realizan un acceso no alineado, y un acceso a una dirección fuera de rango. En ambos casos la ejecución salta a la rutina Data Abort (bucle infinito).
* Undef: en la memoria de instrucciones se incluye un banco de pruebas en el que se ejecuta una instrucción con un código de instrucción desconocido. La ejecución salta a la rutina UNDEF (bucle infinito).

RF2 retorno a modo usuario RTE

**Descripción:** Al terminar la excepción se vuelve a la ejecución original con la instrucción RTE

¿**Cómo**? Explicar brevemente la gestión del PC en la instrucción RTE.

**Verificación:** Sólo se ha realizado para IRQs.Para el resto de excepciones se asume que es un fallo irresoluble y se debe resetear el sistema, pero la gestión es idéntica que para las IRQs. En el banco de pruebas test\_IRQ se puede comprobar que se retorna a la ejecución sin alterarla siempre que se siga un esquema de prólogo y epílogo apropiado (guardando y restaurando los registros en pila). Para ello se utiliza el registro 31 como SP.

RF3 WRO

**Descripción:** la instrucción WRO permite escribir el contenido de un registro en la salida del MIPs

¿**Cómo**? Explicar brevemente la gestión del WRO.

**Verificación:** En el banco de pruebas test\_IRQ se realizan dos WROs (WRO R31, WRO R2).

RF4 Anticipación de operandos

**Descripción:** El procesador es capaz de anticipar los operandos para las instrucciones LW, SW y ARIT a distancia 1 (siempre que el dato a anticipar no venga de un lw) y 2.

¿**Cómo**? Explicar brevemente la gestión de anticipación de operandos.

**Verificación:**

* En el banco de pruebas test\_IRQ se realizan las siguientes anticipaciones:
  + De LW R1, 8(R0) a ADD r31, R1, R31: anticipación de Rs distancia 2 (tras detención de un ciclo del ADD)
  + De SUB r31, R31,R1 a LW R1, 0(R31): anticipación de Rs distancia 1
* Completar con otros casos representativos.

RF5 Riesgos de datos

**Descripción:** El procesador es capaz de detener la ejecución para evitar los riesgos causados por las dependencias en instrucciones lw-uso (distancia 1), beq o WRO (distancias 1 o 2).

¿**Cómo**? Explicar brevemente la gestión de detenciones.

**Verificación:**

* En el banco de pruebas test\_IRQ se realizan las siguientes detenciones:
  + Test1: 1 ciclo de detención por dependencia a distancia 2 entre LW R31, 0(R0) y WRO R31.
  + Test 2: 2 ciclos de detención por dependencia a distancia 1 entre ADD R1, R1, R1 y beq R1, R1, main.
  + Test 3: 1 ciclo de detención por dependencia lw-uso en varios casos. Por ejemplo: de LW R1, 8(R0) a ADD r31, R1, R31.
  + Test 4: 2 ciclos de detención por dependencia a distancia 1 entre ADD R2, R1, R2 y WRO R2.
* Completar con otros casos representativos.

RF6Riesgos de control

**Descripción:** El procesador es capaz eliminar los riesgos de control causados por las instrucciones de salto: BEQ y RTE.

¿**Cómo**? Explicar brevemente la gestión de riesgos de control.

**Verificación:**

* En el banco de pruebas test\_IRQ se realizan varios saltos tomados en los que sólo se ejecutan las instrucciones adecuadas:
  + beq R1, R1, main
  + rte
* verificar un ejemplo en el que no se salte.

RF7 contadores

**Descripción:** los contadores nos dan información de la ejecución del código

¿**Cómo**? Explicar brevemente la gestión de los contadores.

**Verificación:** En el banco de pruebas test\_IRQ se ha comprobado que:

* Por cada instrucción válida que hace su etapa WB se incrementa el contador de instrucciones Ins. Si la instrucción no es válida no se incrementa.
* Data\_stalls contabilizan bien los ciclos debidos a los riesgos de datos incrementando el número indicado en los test mencionados en el RF5.
* Control\_stalls contabilizan bien los ciclos debidos a los riesgos de control, incrementando un ciclo en cada salto tomado (ver pruebas en el RF6).
* Exception\_accepted contabiliza las excepciones aceptadas. Su cuenta coincide con la cuenta que realiza el propio código.
* Exception\_cycles se actualiza cada ciclo en el que el procesador está en modo excepción.

## Cuantificación de horas dedicadas

*Indicar horas por tarea (y persona en el caso de parejas)*

* Estudio del MIPS, VHDL, entorno, instalación…:
* Adición de las nuevas instrucciones:
* Gestión de excepciones:
* Gestión de riesgos:
* Depuración, verificación y programas de prueba:
* Memoria:

## Conclusiones y Autoevaluación

*¿Para qué ha servido vuestro trabajo?, ¿Qué os ha parecido?, ¿qué calificación os otorgáis a vosotros mismos?*