

# Plan de pruebas del modelado de protocolo MESI y MOESI (App)

v. 2023-1

Yordi Brenes, Fátima Leiva, Brian Wagemans,  
Andrey Zúñiga

October 7, 2023



## Introduction

El objetivo de este plan de pruebas es verificar el funcionamiento correcto de los protocolos de coherencia de caché MESI y MOESI. Para esto se realizarán pruebas unitarias de cada transición de ambos protocolos. Además, se harán pruebas funcionales para garantizar que la aplicación cumple con todas las funciones

# Contents

<b>1</b>	<b>Objetivos de las pruebas</b>	<b>3</b>
1.1	Estrategias de prueba . . . . .	3
1.2	Ejecución de pruebas unitarias . . . . .	3
1.2.1	Pruebas MESI . . . . .	3
1.3	Pruebas funcionales . . . . .	4
1.3.1	Generación de código . . . . .	4
1.3.2	Correr código de forma correcta . . . . .	4
1.3.3	Correr código sin generar código . . . . .	4
1.3.4	Correr código de usando step . . . . .	4
1.3.5	Selección de protocolo MESI . . . . .	5
1.3.6	Selección de protocolo MOESI . . . . .	5
1.3.7	Generar reporte . . . . .	5

# 1 Objetivos de las pruebas

1. Verificar que cada transición de estados para una línea de caché en el protocolo MESI y MOESI se produzca correctamente.
2. Validar que se genere el código de pruebas, se corra la aplicación y se escriba en las memorias correctamente.

## 1.1 Estrategias de prueba

Se realizarán dos tipos de pruebas diferentes, funcionales y unitarias. Las pruebas unitarias validarán la transición entre estados de caché. Mientras que las pruebas funcionales probarán las funcionalidades generales del sistema, es decir

1. Generación de código desde interfaz de usuario
2. Escritura en memorias de forma adecuada
3. Selección de protocolo
4. Generación de reporte

Para las pruebas unitarias se utilizará la herramienta xUnit, mientras que la validación de pruebas funcionales se hará de forma manual utilizando la aplicación web.

## 1.2 Ejecución de pruebas unitarias

### 1.2.1 Pruebas MESI

Los posibles estados de MESI son

1. Modificado
2. Exclusivo
3. Shared
4. Invalido

Se realizará una prueba unitaria para cada posible transición de la máquina de estados

#### 1.2.1.1 $M \rightarrow M$ :

Condición:

- Línea en caché en estado M.
- CPU dueño lee la línea.

Resultado esperado: Línea se mantiene en estado M.

Condición:

- Línea en caché en estado M
- CPU dueño escribe en la línea

Resultado esperado: Línea se mantiene en estado M

#### 1.2.1.2 $M \rightarrow S$ :

Condición:

Linea en caché  $S$  con dato  $d$  en estado  $M$

Otro CPU genera un miss al intentar leer  $d$

Resultado esperado: Linea en caché  $S$  con dato  $d$  pasa a estado  $S$   
(Continua...)

### 1.3 Pruebas funcionales

#### 1.3.1 Generación de código

- En la interfaz, presione el botón "Generar código"
- En los cuadros que representan los procesadores aparecerán las primeras 4 líneas de código de cada procesador.

#### 1.3.2 Correr código de forma correcta

- Una vez generado el código.
- Presione el botón de "Correr".
- Ahora las animaciones del bus de datos deberán comenzar.
- Los valores y estados de la memoria compartida y caché deberían cambiar.

#### 1.3.3 Correr código sin generar código

- Con los procesadores vacíos, presione el botón de "Correr".
- Un mensaje se desplegará diciendo que no existen instrucciones que generar.

#### 1.3.4 Correr código de usando step

- En la interfaz gráfica, presione "Generar Código".
- Presione el botón de "Step" de un procesador.
- Ahora deberá de ejecutarse la instrucción del procesador seleccionado.
- Los valores y estados de la memoria compartida y caché deberían cambiar.
- Si presiona "Step" de nuevo, otra instrucción debería ejecutarse.

### **1.3.5 Selección de protocolo MESI**

- En la interfaz gráfica, seleccione el modo MESI
- Genere código aleatorio presionando "Generar Código"
- A continuación, presione "Correr"
- Durante toda la ejecución, ningún estado será O en ningún momento de la ejecución
- Repita los pasos anteriores una vez más para asegurar los resultados

### **1.3.6 Selección de protocolo MOESI**

- En la interfaz gráfica, seleccione el modo MOESI.
- Genere código aleatorio presionando "Generar Código".
- A continuación, presione "Correr".
- Durante toda la ejecución, en cualquier momento alguna línea de caché está en estado O.
- Repita los pasos anteriores una vez más para asegurar los resultados.

### **1.3.7 Generar reporte**

- En la interfaz gráfica, presione el botón "Generar Código"
- Presione el botón de "Correr".
- Al acabar de correr todo el código, se desplegará un reporte en pantalla.
- Al acabar de correr todo el código, se desplegará un reporte en pantalla.
- El reporte debe desplegar transacciones totales, read requests, write requests e invalidaciones