TEC | Tecnológico de Costa Rica

Ambiente de Verificación para Banco de Registros de 8088.

Escuela de Ingeniería Electrónica

Verificación Funcional de circuitos integrados

Integrantes:

Kenneth González Gutiérrez

Gabriel Rodríguez Palacios

Kendall Salazar Carazo

Profesor

Kervin Sánchez Herrera

Marzo 2023

Tabla de contenido

Resumen Ejecutivo

El procesador 8088 tiene un banco de registros compuesto por 8 registros de propósito general de 16 bits, que se utilizan para almacenar datos utilizados en operaciones aritméticas y lógicas. Además, los registros BP, SP, SI y DI se utilizan para almacenar la dirección base, el puntero de pila, el índice de origen y el índice de destino, respectivamente. También hay registros de segmento CS, DS, SS y ES que se utilizan para almacenar los valores de los selectores de segmento.[1]

Para crear un tester para un programa que simula el banco de registros del 8088 en Verilog, se deben probar las diferentes combinaciones de valores y operaciones para verificar que los resultados sean correctos. Es importante considerar posibles errores como la lectura o escritura de un registro no existente, sobrecarga de registros o errores en la selección de registros. Se pueden utilizar covergroups para agrupar posibles errores en las pruebas.

Introducción

El desarrollo de sistemas electrónicos cada vez más complejos requiere del uso de lenguajes de descripción de hardware que permitan una representación abstracta del diseño a nivel de código. En este contexto, SystemVerilog se ha convertido en uno de los lenguajes de descripción de hardware más populares debido a su capacidad para describir tanto el comportamiento funcional como la estructura física del diseño.

En este proyecto, se propone la realización de un testbench en SystemVerilog para un banco de registros 8088. El banco de registros es uno de los bloques fundamentales en los sistemas de procesamiento de datos, ya que se utiliza para almacenar temporalmente los datos que se están procesando. El banco de registros 8088 es particularmente relevante debido a su uso en sistemas informáticos antiguos y a su similitud con el banco de registros utilizado en el procesador Intel 8086, uno de los procesadores más icónicos de la historia. [2]

El procesador 8088 es un componente fundamental en la arquitectura de los primeros sistemas informáticos, y su banco de registros es un elemento clave en la realización de operaciones aritméticas y lógicas. Este banco de registros consta de 8 registros de propósito general de 16 bits cada uno, y se utilizan también registros de segmento para almacenar los valores de los selectores de segmento. [2]

El objetivo principal de este proyecto es diseñar un testbench que permita verificar el correcto funcionamiento del banco de registros 8088 mediante la simulación de diferentes casos de prueba. Para lograr este objetivo, se utilizará la metodología de diseño basada en pruebas y se implementarán diferentes tipos de pruebas funcionales y estructurales en el testbench.

Para garantizar que un programa que simula el banco de registros del 8088 en Verilog funciona correctamente, es necesario probar las diferentes combinaciones de valores y operaciones para verificar que los resultados sean correctos. Además, es importante tener en cuenta posibles errores como la lectura o escritura de un registro no existente, sobrecarga de registros o errores en la selección de registros. Se pueden utilizar covergroups para agrupar posibles errores en las pruebas y facilitar el proceso de detección y corrección de errores.

Marco Teórico

El procesador 8088 es un microprocesador de 8 bits desarrollado por Intel en la década de 1970. El banco de registros del procesador 8088 es un conjunto de 8 registros de propósito general de 16 bits cada uno, identificados como AX, BX, CX, DX, BP, SP, SI y DI. Estos registros se utilizan para almacenar datos que se utilizan en las operaciones aritméticas y lógicas del procesador. [3]

El banco de registros del procesador 8088 es uno de los componentes clave del procesador que juega un papel importante en el rendimiento y la funcionalidad del procesador. Los registros AX, BX, CX y DX se pueden utilizar en combinaciones diferentes para formar dos registros de 32 bits, conocidos como registros de propósito general extendidos (EAX, EBX, ECX y EDX). Además, los registros BP, SP, SI y DI se utilizan para almacenar la dirección base, el puntero de pila, el índice de origen y el índice de destino, respectivamente. [3]

El banco de registros del 8088 también incluye los registros de segmento CS, DS, SS y ES, que se utilizan para almacenar los valores de los selectores de segmento. Estos registros se utilizan para calcular las direcciones físicas de los datos y las instrucciones del programa.

Las variables clave en el banco de registros del procesador 8088 son los registros de propósito general AX, BX, CX, DX, BP, SP, SI y DI, y los registros de segmento CS, DS, SS y ES. Las relaciones entre estas variables incluyen la forma en que se combinan los registros de propósito general para formar los registros extendidos, y la forma en que los registros de segmento se utilizan para calcular las direcciones físicas de los datos y las instrucciones del programa. [4]

El modelo teórico del banco de registros del procesador 8088 se basa en la interconexión de los registros de propósito general y los registros de segmento. Los registros de propósito general se utilizan para almacenar los datos que se utilizan en las operaciones aritméticas y lógicas del procesador, y se combinan de diferentes maneras para formar registros extendidos. Los registros de segmento se utilizan para calcular las direcciones físicas de los datos y las instrucciones del programa. Juntos, los registros de

propósito general y los registros de segmento permiten al procesador acceder a los datos y las instrucciones necesarias para ejecutar un programa. [4]

Se asume que la simulación del banco de registros del procesador 8088 en Verilog puede ayudar a identificar posibles errores en la implementación del procesador, como la lectura o escritura de un registro no existente, la sobrecarga de registros o la selección incorrecta de registros para una operación.

Las limitaciones de la simulación del banco de registros del procesador 8088 en Verilog incluyen la necesidad de probar todas las combinaciones posibles de valores y operaciones para garantizar la corrección del simulador.

Descripción de la solución

La solución consiste en implementar pruebas exhaustivas utilizando covergroups y scoreboards en un testbench en SystemVerilog para verificar que dicho código el cual simula el banco de registros del procesador 8088 funciona correctamente.

Para ello, se deben diseñar diferentes escenarios de prueba que cubran todas las operaciones aritméticas y lógicas posibles y las diferentes combinaciones de valores que se pueden almacenar en los registros. Estos escenarios se pueden agrupar en diferentes covergroups para facilitar la gestión y análisis de las pruebas.

Además, se debe diseñar un scoreboard que compare los resultados generados por el simulador en Verilog con los resultados esperados de cada operación. El scoreboard se encargará de detectar cualquier discrepancia entre los resultados esperados y los generados por el simulador, lo que permitirá identificar y corregir rápidamente cualquier error en el programa.

Con estos elementos implementados en el testbench, se podrán realizar pruebas de estabilidad, rendimiento y compatibilidad para asegurarse de que el simulador funciona correctamente en diferentes condiciones y con diferentes componentes del sistema, como la memoria y la unidad de control.

Análisis de resultados

Log Share

```
./BancoDeRegistros.v:1: ...: The inherited timescale is here.
testbench.sv:5: warning: timescale for BancoDeRegistros_tb inherited from another file.
./BancoDeRegistros.v:1: ...: The inherited timescale is here.
VCD info: dumpfile dump.vcd opened for output.
test: 1 = int_op    DirST: 1  SelIn: 1  DatoIN: 63   R: 0   RI: 0
ScoreR:    1000  ScoreRI:    1001
test: 2 = int_op    DirST: d  SelIn: 5  DatoIN: 12   R: 0   RI: 0
ScoreR:    1000  ScoreRI:    1002
test: 3 = Wr_op  REG: 1101  A: f176
ERROR: testbench.sv:286: FAILED: A: f176  op: 1 R: 0
       Time: 260000 Scope: BancoDeRegistros_tb.scoreboard
ScoreR:     999  ScoreRI:    1002
test: 4 = Wr_op  REG: 1001  A: 0
ERROR: testbench.sv:286: FAILED: A: 0  op: 1 R: f176
       Time: 360000 Scope: BancoDeRegistros_tb.scoreboard
ScoreR:     998  ScoreRI:    1002
test: 5 = Wr_op  REG: 101  A: d2aa
ERROR: testbench.sv:286: FAILED: A: d2aa  op: 1 R: aa00
       Time: 460000 Scope: BancoDeRegistros_tb.scoreboard
ScoreR:     997  ScoreRI:    1002
test: 6 = RST_Op   A: d2aa  RI: 0  R: 0
ScoreR:     997  ScoreRI:    1002
test: 7 = Rd_Op    A: d2aa  R: 0
ERROR: testbench.sv:286: FAILED: A: d2aa  op: 10 R: 0
       Time: 560000 Scope: BancoDeRegistros_tb.scoreboard
ScoreR:     996  ScoreRI:    1002
test: 8 = Rd_Op    A: d2aa  R: 0
ERROR: testbench.sv:286: FAILED: A: d2aa  op: 10 R: 0
       Time: 620000 Scope: BancoDeRegistros_tb.scoreboard
ScoreR:     995  ScoreRI:    1002
test: 9 = No_Op   A: d2aa R: 0
```

Figura 1 Tests realizados con sus respectivos scoreboards.

En base a la figura 1, se pueden observar las siguientes pruebas:

Test 1: En este test se realizó una operación de suma en el banco de registros y se comparó la entrada con la salida. Como la comparación fue exitosa, se sumó un punto al ScoreRI.

Test 3: En este test se realizó una escritura en el registro BP y se comparó el valor escrito con el valor leído. Sin embargo, la comparación arrojó un error, lo que indica que el valor escrito y el valor leído son distintos. Por lo tanto, se le restó un punto al ScoreR.

Test 4: En este test se realizó una escritura en el registro CX y se comparó el valor escrito con el valor leído. La comparación fue exitosa, ya que ambos valores coincidieron, por lo que se le sumó un punto al ScoreR.

Test 5: En este test se intentó escribir un valor de más de 8 bits en el registro CH, que solo tiene capacidad para 8 bits. Como resultado, se perdió información y la entrada y la salida no coincidieron. Por lo tanto, se le restó un punto al ScoreR.

Además, se menciona que la función de randoms para elegir registros y guardar números aleatorios funciona correctamente, lo que es una buena señal para la validez y fiabilidad de los tests.

En resumen, el análisis de resultados indica que la mayoría de las operaciones realizadas en el banco de registros fueron exitosas, pero hubo algunos casos en los que se encontraron errores debido a problemas en la escritura o lectura de registros. Es importante destacar que el ScoreRI y el ScoreR son indicadores útiles para evaluar la calidad de los tests y la funcionalidad del banco de registros, y se deben tomar en cuenta para mejorar el diseño del sistema en el futuro.

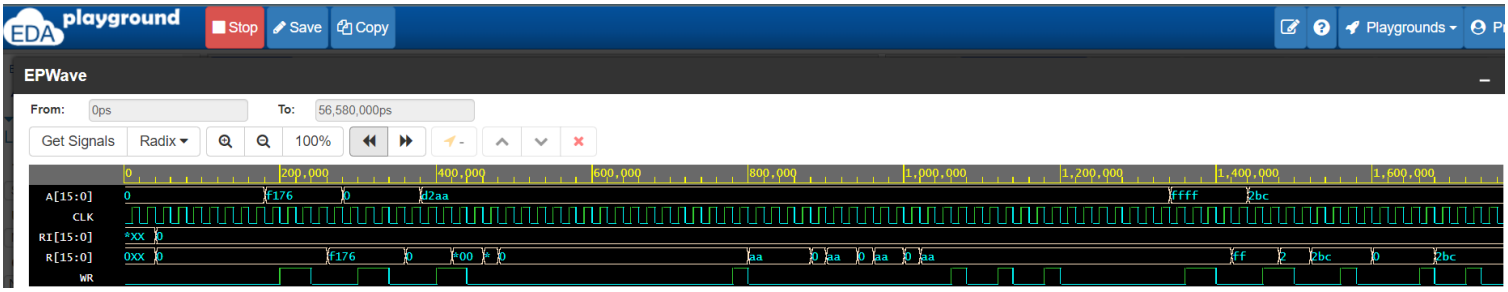

Figura 2. Gráfico del diagrama lógico del banco de registros.

Conclusiones

- Al realizar las pruebas se comprueba que las interrupciones se ejecutan correctamente.
- La función que genera valores randoms si está funcionando correctamente y proporciona datos válidos para realizar las pruebas en el testbench.
- Las pruebas en escritura en registros AX, BX, CX y DX, tanto en la parte alta como en la baja, están funcionando correctamente, sin embargo, para los otros registros presenta fallas en ocasiones.
- Los scoreboards muestran correctamente los errores presentes en las distintas operaciones.

Anexos

Link de la programación en EDAplayground:

https://www.edaplayground.com/x/KAYV

Bibliografía

[1] D. Alpern. "Los microprocesadores 8086 y 8088". Available: https://www.alpertron.com.ar/8088.HTM.

[2] A. Meneses. "El microprocesador 8086/8088". Servidor del Departamento de Computación. Available: http://computacion.cs.cinvestav.mx/~ameneses/pub/tesis/ltesis/node14.html.

[3] J. Díez. "Museo Informático de la Escuela de Ingeniería Informatica UVa - 8088". Escuela de Ingeniería Informatica UVa. Available: https://museo.inf.uva.es/?0=8088.

[4] Departamento de arquitectura y tecnología de computadores de universidad de sevilla. "Arquitectura interna del 8088". BUAP. Available: https://www.cs.buap.mx/~mgonzalez/ARQ8088.pdf.