Вторая лабораторная работа по Архитектуре ЭВМ

Сайфуллин Искандар M3136

11 ноября 2023

1 Версии програм и репозиторий

`Logisim Evolution` — 3.8.0
`Icarus Verilog` — 12.0
Ссылка на репозиторий — https://github.com/skkv-itmo2/itmo-comp-arch-2023-circuit-1sSay

2 Логический уровень

2.1 Модули

Я разбил стек на множество модулей. Вот главные из них:

1. `Counter` — счётчик по модулю 5. Хранит в себе трёхбитное число, которое может быть увеличено или уменьшено на единицу.
2. `Memory` — здесь по большей части реализована вся логика стека. Там хранятся все ячейки памяти стека. Кстати, все ячейки реализованы с помощью D-триггеров, которые работают по фронту. (А они в свою очередь на D-триггере с высоким уровней синхронизации. А они в свою очередь на RS-триггере с синхронизацией тоже по высокому уровню)
3. `InOut` — модуль для реализации входовыхода. По факту там стоят вентили, который пропускает сигнал, если на `Clock` и `Pop` или `Get`.

Также есть всякие декодеры, сумматоры и другие утилитарные модули, которые используются во многих местах.

2.2 Описание работы

Как, собственно, работает стек?

1. `Clock`. С ним всё завязано, пока он выключен, ничто, кроме `Reset` не будет работать.
2. `Reset`. Он обнуляет все ячейки памяти. От синхронизации никак не зависит.
3. `Push`. Сначала получается индекс на нужную ячейку памяти (которая идёт после вершины), а потом данные в неё записываются. Счётчик при этом, разумеется увеличивается на единицу (по модулю 5)
4. `Pop`. Сначала получается индекс на нужную ячейку памяти (индекс вершины), а потом данные из неё считываются. Счётчик при этом, разумеется уменьшается на единицу (по модулю 5). Данные в самой ячейке никуда при это не пропадают, потому что это ни к чему.
5. `Get`. Сначала вычисляется индекс ячейки. Вычитаем одно число из другого (разумеется, по модулю 5) и выводим данные из нужной ячейки. Счётчик при этом никак не изменяется.

3 Структурный уровень

Работает абсолютно также, как и логический. Просто переписан на `Verilog` (Простите за некрасивый код, я разбирался в синтаксисе прямо во время написания лабораторной >_<)

4 Поведенческий уровень

Сначала создаём массив из 4-хбитный регистров для ячеек стека, а также трехбитный регистр для индекса вершины стека.

В модуле завобим `initial`, в котором есть 2 `always`. Первый для уставовки `z` в data при переходе `clock` с единицы в ноль (Это нужно для входо-выходов). Второй для реализации логики стека. В случае, если `Clock` равен единице, то смотрим на значение команды с помощью `case` и выполняем соответсвующую операцию. Также при сбросе асинхронно зануляем все значения ячеек в стеке и счётчик.