

МОДЕЛЬ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ С РЕЗЕРВИРОВАНИЕМ

Целью данной работы было создание модели отказоустойчивой вычислительной системы, основанной на принципе мажорирования. Создание модели позволит исследовать эффективность работы различных алгоритмов синхронизации, изучать временные характеристики вычислительных систем различной конфигурации. Модель может служить основой для создания устройств на языках HDL.

В качестве средства разработки был выбран пакет AnyLogic.

Модель вычислительной системы строится из доменов двух типов. Каждый домен включает в себя 3 экземпляра вычислительных устройств, для доменов первого типа это процессоры, второго типа – память. Процессоры могут формировать сигналы запроса, устройства памяти принимают и обрабатывают запросы и формируют сигналы ответа.

Кроме того, в каждый домен входят устройства согласования, доменные коммутаторы и генераторы ошибок, также представленные в 3 экземплярах. В вычислительный домен вдобавок входит генератор адреса. Каждый домен делится на 3 узла, состоящих из одного устройства каждого типа. Узлы одного домена одновременно выполняют одинаковые задачи, сравнение и синхронизация результатов работы узлов производятся по принципу мажорирования. Устройства согласования и доменные коммутаторы служат для связи узлов между собой. Устройства согласования синхронизируют запросы, поступающие от процессоров и устройств памяти, сравнивают результаты выполнения операций и отключают узлы в случае отказа. Доменные коммутаторы служат для соединения доменов между собой и для скрытия собственных вычислительных устройств от соседних. Связь между доменами реализуется с помощью шины, выполненной в форме внешнего устройства, включающего в себя все домены. Обмен сигналами производится в режиме опроса готовности.

Структурная схема модели вычислительной системы представлена на рис. 1.

Генератор адреса и генератор ошибок представляют собой псевдоустройства, обеспечивающие корректное функционирование модели. Первый служит только для случайного выбора домена памяти, к которому произойдёт следующее обращение. Второй позволяет имитировать отказы в работе вычислительной системы и анализировать её поведение в различных ситуациях.

Модель работает по следующему алгоритму:

1. Вычислительное устройство подаёт запрос.
2. Устройства согласования ожидают запросы от всех вычислительных устройств, в случае отсутствия запроса от одного из устройств на протяжении определённого времени, узел считается отказавшим.
3. Устройства согласования подают сигналы разрешения вывода на доменные коммутаторы.
4. Доменные коммутаторы подают сигналы на порты вывода.
5. Генератор адреса вычисляет номер домена памяти, которому будет передан запрос.
6. При следующем опросе портов вывода вычислительных доменов происходит соединение с запрашиваемым доменом памяти в том случае, если он свободен, иначе вычислительный домен блокируется до освобождения домена памяти.
7. Доменные коммутаторы домена памяти сравнивают входные сигналы, при корректных входных сигналах устройствам согласования передаётся сигнал, разрешающий работу вычислительного устройства.
8. Устройства согласования передают сигнал запроса устройствам памяти.
9. Устройства памяти «обрабатывают» запрос и формируют сигнал ответа.
10. Сигнал ответа передаётся на порты вывода домена памяти аналогично шагам 2-4.
11. Сигнал ответа передаётся через вычислительный домен аналогично шагам 7-9.

12. Домены разъединяются.

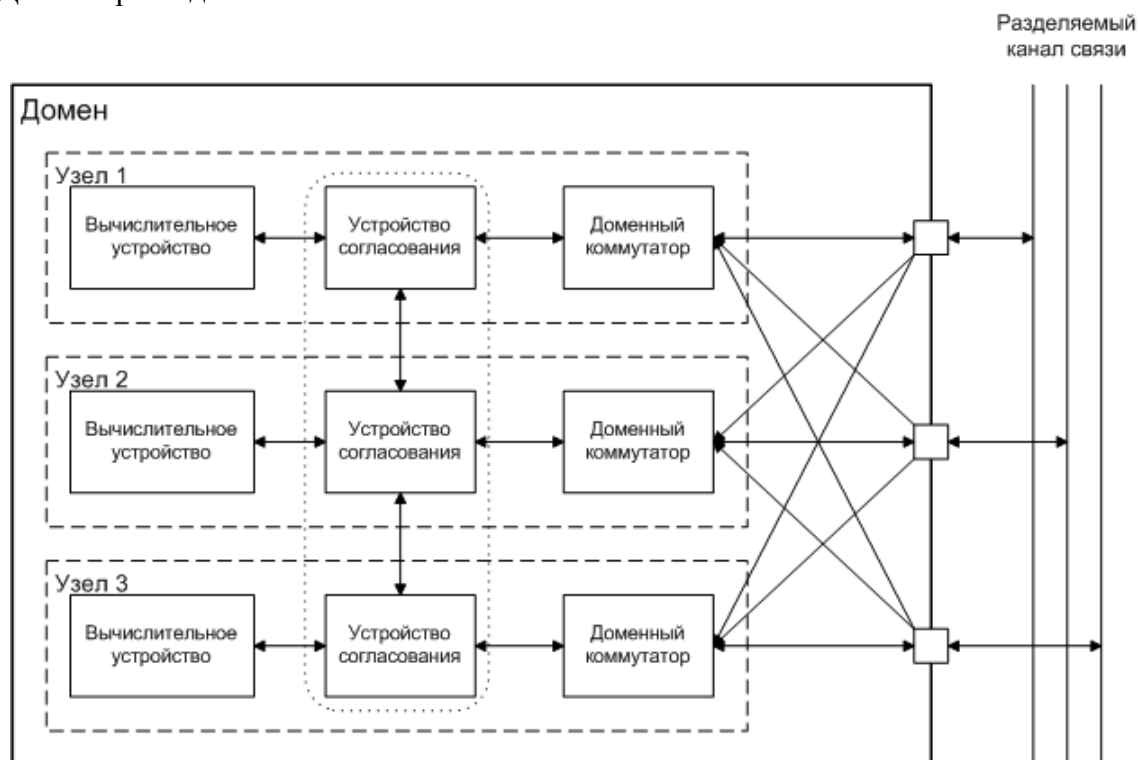


Рис. 1. Структурная схема модели вычислительной системы

В результате работы была получена модель, позволяющая задавать конфигурацию вычислительной системы (количество доменов каждого типа), временные параметры работы вычислительных устройств, вероятность происхождения отказов. Время происхождения всех событий фиксируется, что позволяет анализировать временные характеристики работы вычислительной системы заданной конфигурации.

В дальнейшем возможны следующие варианты усовершенствования модели:

- Усложнение алгоритмов передачи, обработки и упорядочивания запросов, организация очередей запросов.
- Замена разделяемого канала связи на троированную системную шину с разделением доступа, создание устройства-арбитра.
- Усложнение алгоритмов выявления и обработки отказов.