Повышение надежности с использованием реконфигурации Multiclet R1

Зырянов Б.А., д.т.н, Стрельцов Н.В.

По сравнению с традиционными процессорами, мультиклеточные процессоры имеют ряд отличий. Из них, с точки зрения построения надёжных систем, наиболее значимы два. А именно, динамическая реконфигурация и двухуровневая организация программ.

1. Динамическая реконфигурация мультиклеточного процессора позволяет использовать клетки как отдельные процессорные устройства, либо группу клеток как отдельный мультиклеточный процессор (***task***-группу). Например, четыре клетки Multiclet R1 могут образовывать четыре ***task***-группы по одной клетке, либо две ***task***-группы по две клетки, либо две ***task***-группы, одна из которых состоит из одной клетки, а вторая из трех клеток, либо одну ***task***-группу из четырех клеток. Каждая клетка может входить в любую ***task***-группу, но только в одну. Состав ***task***-групп в процессе решения задач может меняться динамически. При этом исполняемый код абсолютно независим от состава ***task***-группы, которая его выполняет. Так один и тот же экземпляр исполняемого кода может одновременно исполняться двумя ***task***-группами, например, из одной и трех клеток.
2. Программа мультиклеточного процессора состоит из параграфов, которые представляют собой одну или несколько информационно замкнутых и частично-упорядоченных последовательностей команд. Команда мультиклеточного процессора в качестве операндов может использовать либо адреса переменных, либо номера GPR, либо ссылки на результаты предшествующих команд (частичная упорядоченность). При этом команды могут ссылаться только на результаты команд своего параграфа (информационная замкнутость). Ссылка за границы параграфа принципиально недопустима. Она, как правило, приводит к «зависанию» ***task***-группы. Невозможно получить для выполнения *i*-той команды в ***task***-группе результат *(i-j)*-той команды, где $j>i$ и которого нет в принципе, а невыполнение одной команды, через некоторое время полностью блокирует выборку команд всей ***task***-группы.

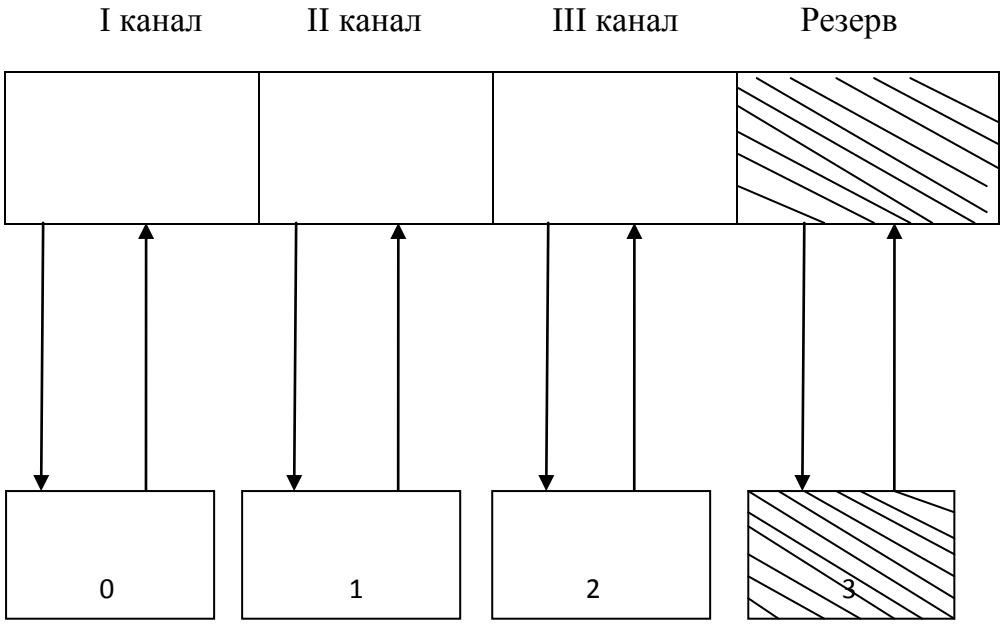
Динамическая реконфигурация позволяет организовать одновременное выполнение одного и того же исполняемого функционального кода разными ***task***-группами. При этом каждая ***task***-группа может иметь свой экземпляр этого кода, что позволит избежать значительного количества конфликтов при обращении к памяти программ и обеспечить равное или близкое время его исполнения. Очевидно также, что процесс выполнения может быть организован поэтапно и результаты каждого этапа каждая ***task***-группа может записывать в свою зону

памяти. Выполнив очередной этап, ***task***-группа используя традиционные семафоры формирует в памяти признак о завершении работ и приостанавливается.

Контроль полученных результатов может быть также организован с использованием ***task***-групп, этих или других, которые включаются тогда, когда сформированы признаки завершения очередного этапа. При этом каждая ***task***-группа должна иметь свой физический канал доступа к памяти других ***task***-групп.

Фактически, формирование ***task***-групп – это создание в рамках одной аппаратной среды многомашинной системы. Сформировать подобную многомашинную систему в рамках традиционных многоядерных процессоров – невозможно так как они имеют один, физически не разделяемый, общий ресурс – память с одним каналом доступа.

Рассмотрим одну из возможных схем работы подобной системы. Предположим, что три клетки (три ***task***-группы) одновременно выполняют функциональный код, а одна клетка (***task***-группа) находится в резерве.

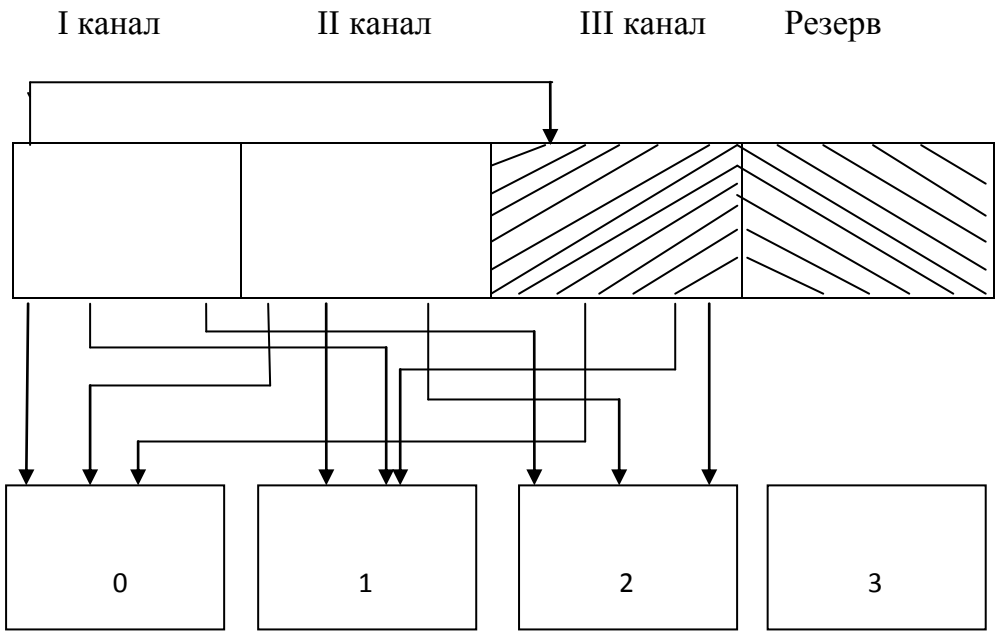


Штатная работа

Каждая клетка, выполнив очередной этап и сформировав признак о его завершении, анализирует наличие аналогичных признаков от соседних клеток. Если их нет, то она переходит в режим ожидания. Длительность этого состояния может быть задана либо таймером, либо счетчиком цикла с постоянным опросом данных признаков. По завершению периода ожидания каждая клетка осуществляет анализ признаков. Возможны следующие случаи.

1. Все три клетки сформировали признаки завершения этапа. В этом случае они одновременно сравнивают результаты работы друг друга и

записывают в память результаты сравнения с формирование признака завершении этого шага. При появлении всех трех данных признаков клетки проводят оценку результатов сравнения. Если оценки совпали, то очередной этап считается выполненным, и клетки переходят к следующему этапу. Если оценки не совпали, а именно результаты одной клетки в которой произошел сбой, отличаются от результатов двух других, то одна из этих двух клеток (например, с минимальным физическим номером) копирует содержимое своей области памяти в область памяти сбойной клетки. При этом она подсчитывает контрольную сумму области памяти сбойной клетки и сравнивает ее с контрольной суммой области памяти другой исправной клетки. Если контрольные суммы совпали, то система считается восстановленной и клетки переходят к следующему этапу.

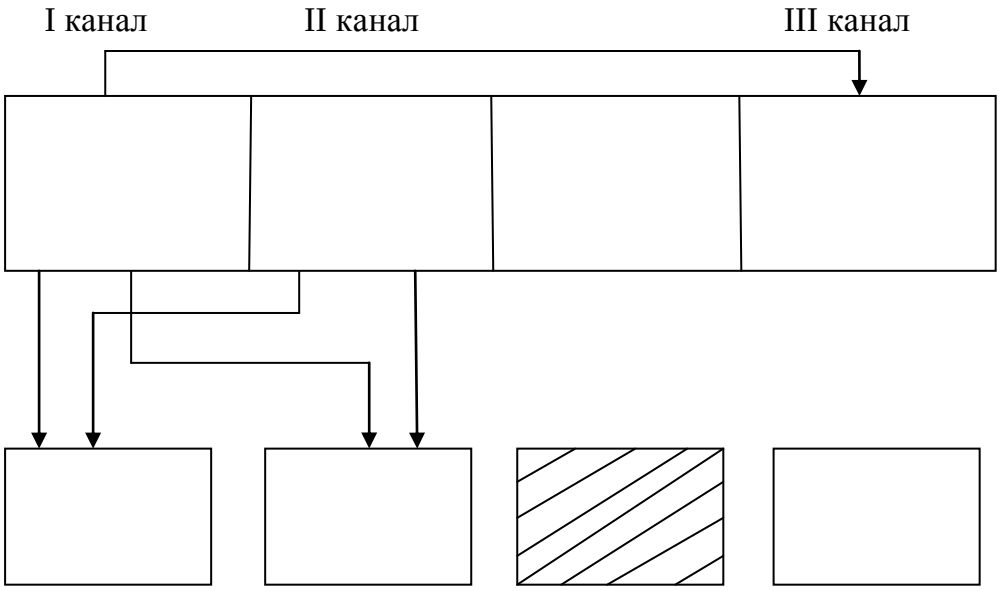


Сравнение результатов и восстановление сбойного канала

Если контрольные суммы не совпали, то процесс повторяется. После третьего раза фиксируется отказ системы.

2. Только две клетки сформировали признаки завершения этапа. В этом случае фиксируется отказ одной клетки и если в результате сравнения оставшихся двух клеток фиксируется их несовпадение, то считается, что система отказала. Если результаты совпали, то в область памяти резервной клетки переписываются с контролем содержимое области памяти исправной клетки с минимальным физическим номером. Результат контроля также сравнивается с контрольной суммой области памяти другой исправной клетки. Если они совпадают, то резервная

клетка включается в работу и все три клетки переходят к выполнению следующего этапа. Иначе фиксируется отказ системы.



Отказ канала и подключение резерва

3. Если только одна клетка сформировала признак завершения этапа, то фиксируется отказ системы.

Подобная организация работы не является полной заменой традиционных средств повышения надежности, поскольку критична к частоте сбоев и не учитывает сбои, приводящие к полной, но кратковременной потере работоспособности. Тем не менее, она может служить вполне работоспособной альтернативой, не использующей дорогостоящие методы, либо дополнительным, наряду с уже использующими, средством повышения надежности.