Оглавление

Введение.

СЧЕТ ПЕРЕХОДОВ.

КОДЫ ЦИКЛИЧЕСКОГО ИЗБЫТОЧНОГО КОНТРОЛЯ.

СИГНАТУРНЫЙ АНАЛИЗ.

КАКИЕ ИЗДЕЛИЯ ПРИГОДНЫ ДЛЯ ИСПЫТАНИЯ МЕТОДОМ СИГНАТУРНОГО АНАЛИЗА.

ПРОСТОЙ СИГНАТУРНЫЙ АНАЛИЗАТОР.

Структурная схема простого сигнатурного анализатора.

Секция анализа простого сигнатурного анализатора.

Схема индикации сигнатуры зафиксированной в анализаторе.

Описание элементной базы используемой для создания простого сигнатурного анализатора.

ТЕСТИРОВАНИЕ В РЕЖИМЕ СВОБОДНОГО СЧЕТА С ИСПОЛЬЗОВАНИЕМ СИГНАТУРНОГО АНАЛИЗАТОРА.

Системное ядро. Режим свободного счета.

Использование сигнатурного анализатора в режиме свободного счета.

ТЕСТИРОВАНИЕ ПЗУ В РЕЖИМЕ СВОБОДНОГО СЧЕТА.

ТЕСТ–ЦИКЛЫ СИГНАТУРНОГО АНАЛИЗА.

ВСТРОЕННЫЕ СРЕДСТВА ДЛЯ СИГНАТУРНОГО АНАЛИЗА

ОГРАНИЧЕНИЯ СИГНАТУРНОГО АНАЛИЗАТОРА

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗУЕМОЙ И РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ:

#

# Введение.

 Аналоговые схемы проверяются путем подачи известных тест-наборов и контроля каждого узла в тракте распространения сигнала с помощью осциллографа. При обнаружении нестандартного сигнала подозрение падает на ту часть схемы, которая управляет узлом, и именно она исследуется более тщательно. В принципе таким же образом можно проверять и цифровые схемы: подавать известные тест-наборы и контролировать каждый узел в тракте распространения сигнала. Однако цифровые системы радикально отличаются от аналоговых систем не только самой природой сигналов, но и наличием гораздо большего числа сигнальных входов. Если бы цифровой системой можно было управлять таким образом, что на всех ее входах действовал четко определенный тест-набор, то каждый узел можно было бы проверить на фиксированный набор событий. В любом узле при каждом выполнении стимулирующей программы возникает один и тот же набор, и его можно использовать для проверки правильной работы узла. Если измеряемый набор отличается от ожидаемого, следует подозревать наличие отказа между данным узлом и стимулируемыми точками системы. При заданном фиксированном тест-наборе измеряемый набор в любом узле оказывается уникальным, и его можно использовать для контроля по принципу “проходит/не проходит”. Измеряемая реакция узла на известные тест-набор называется “сигнатурой” (т.е. подписью) по аналогии с уникальностью рукописных подписей у людей. Рассмотренный принцип лежит в основе целой области цифрового тестирования, называемой *сигнатурным анализом*.

 Сигнатурный анализ разработан на основе двух применявшихся ранее способов контроля ошибок; первым является способ проверки логических узлов в цифровой системе, называемой *счетом переходов*, а вторым – способ контроля ошибок, называемый *циклическим избыточным контролем*. Сигнатурный анализ как бы моделирует первый способ, но действует аналогично второму способу.

# Счет переходов.

 Если узел стимулировать известным и периодическим тест-набором, то сигнал в узле можно считать псевдослучайной двоичной последовательностью, которая на коротких интервалах выглядит случайной, но повторяется при выборе большего масштаба времени.

 Широко распространенный и простой способ контроля заключается в том, чтобы подсчитать переходы сигнала из одного состояния в другое и использовать полученное число в качестве идентификатора узла. Общее число изменений состояния стимулируемого узла может быть очень большим, поэтому необходим какой-то способ сжатия информации. Обязательным условием счета переходов является определенное “временное окно”, в течении которого подсчитываются переходы в узле. В интервале “временного окна” выполняется тест-программа, которая стимулирует проверяемый узел и по возможности должна быть периодической для удобства воспроизведения и сравнения результатов нескольких измерений. В конкретной системе с определенными “временным окном” и тест-последовательностью результаты нескольких подсчетов переходов должны быть идентичными. Для применения метода счета переходов (СП) как способа поиска неисправностей необходимо измерить и задокументировать числа переходов в каждом узле. При возникновении неисправности исследователь выполняет тест-программы, измеряет числа переходов в подозреваемых узлах и сравнивает их с документированными значениями. Любые расхождения свидетельствуют о наличии неисправности, и с помощью систематической процедуры ее можно локализовать.

# Коды циклического избыточного контроля.

 В вычислительной системе, где данные передаются, например с дискового накопителя в основную память, с высокой скоростью, необходимы средства контроля отсутствия ошибок при передачи. Простой контроль с битом четности оказывается недостаточным при передачи блоков данных и применим только в асинхронных последовательных линиях связи, в которых информация передается по символам. Для преодоления ограничений контроля по четности и обеспечения обнаружения одно- и много битных ошибок в быстродействующих синхронных линиях связи были разработаны коды циклического избыточного контроля.

 Если взять 8-битное двоичное слово, его можно рассматривать как коэффициент полинома степени 7. Двоичный набор B7B6B5B4B3B2B1B0 есть сокращенное представление двоичного полинома:

B7\*27 + B6 \*26 + B5\*27+B4\*24+B3\*23+ B2\*22+B1\*21+B0\*20

 Например, двоичное слово 11000001 можно представить в виде:

1\*27 + 1\*26 + 0\*27+0\*24+0\*23+ 0\*22+0\*21+1\*20 .

 Для простоты обозначим переменную в полиноме вместо конкретного значения 2 символом X. Тогда то же самое слово с фиктивной переменной X принимает вид:

A(X)=1\*X7+1\*X6+0\*X5+0\*X4+0\*X3+0\*X2+0\*X1+1\*X0

 Данный полином можно сократить до выражения:

A(X)=X7 +X6 + X0 = X7 + X6 + 1.

 Для понимания принципа циклического избыточного контроля (ЦИК) предположим, что имеются двоичный поток B(X) и порождающий полином G(X). Если разделить полином B(X) на G(X), получится частное Q(X) и остаток R(X):

B(X)=G(X)\*Q(X)+R(X).

 Прибавляя –R(X) к обеим частям уравнения, получим :

B(X) - R(X) = G(X)\*Q(X).

 При способе ЦИК передаваемый двоичный поток делится на порождающий полином и получающийся остаток добавляется в передаваемый двоичный поток. В приемнике входной двоичный поток и остаток делятся на тот же самый полином, что и в передатчике, с образованием остатка. В результате деления на приемной стороне линии связи остаток должен быть равным нулю, так как B(X) + R(X) точно делится на G(X). При получении нулевого остатка на приемной стороне обнаруживаемых ошибок при передачи не возникло.

 При способе ЦИК наиболее широко применяется 16-битный полином вида:

ЦИК-16 = X16 + X15 + X2 + 1

 Для реализации циклического избыточного контроля применяется 16-битный регистр сдвига, на вход которого подается контролируемый двоичный набор B(X), а из соответствующих разрядов регистра сдвига берутся сигналы обратной связи. Входной сигнал и все сигналы обратной связи суммируются по mod 2 (операция исключающего ИЛИ), в результате чего получается линейная последовательностная схема, так как сумматор по mod 2 придает один и тот же вес каждому входному биту. Полином называется характеристическим полиномом генератора, а фактические точки отводов обратной связи находятся из его инверсии. Инверсное (или обратное ) выражение находится путем вычитания каждого члена в характеристическом полиноме из X16 в случае кода ЦИК-16, что дает выражение обратной связи X16 + X14 + X1 + 1. Следовательно, в элемент исключающего ИЛИ подаются входной сигнал и отводы от первого, четырнадцатого и шестнадцатого разрядов регистра сдвига (рис. 1).



Рис. 1. Регистр сдвига с линейной обратной связью

 Регистр сдвига синхронизируется специальными сигналами, и входной поток данных делится на характеристический полином X16 + X15 + X2 + 1. По окончании двоичного потока остаток выдвигается из регистра и добавляется к передаваемому потоку данных. Ранее было показано, что остаток вычитается из двоичного потока [B(X) – R(X)], но в арифметике по mod 2 операции сложения и вычитания дают одинаковые результаты, поэтому передача B(X) + R(X) производит то же самое действие.

 Порождающий полином ЦИК-16 имеет четное число членов, и представим в виде ЦИК-16 = (X+1)\*(X15 + X + 1). Множитель(X+1) включен намеренно, чтобы все необнаруживаемые ошибки имели четный паритет. Чтобы скрыть ошибку, должно возникнуть четное число ошибочных бит, которые обычно группируются. Однако циклический избыточный контроль всегда обнаруживает однобитные ошибки во входном потоке. Таким образом, коды ЦИК представляют собой способ обнаружения в потоке данных всех однобитных ошибок и высокий процент обнаружения всех многобитных ошибок. В связи с этим они широко применяются в схемах быстродействующих цифровых передач, в которых передаются не отдельные символы, а целые блоки данных.

# Сигнатурный анализ.

В циклическом избыточном контроле входной двоичный набор подается в линейную последовательностную схему, которая осуществляет деление двоичного потока на некоторый характеристический полином, и в регистре сдвига образуется остаток от деления. Обычно остаток добавляется к передаваемому двоичному потоку в качестве кода, обнаруживающего ошибки. Если же вместо добавления остатка к двоичному потоку вывести его на индикацию, это значение будет уникальным для входного двоичного набора. Имея запоминающие элементы, схема учитывает все прошлые и текущие события и может обрабатывать очень длинные потоки данных. Уникальный остаток для конкретного входа служит как бы “отпечатками пальцев” этого набора и может использоваться для его идентификации. Зависимость остатка от входного двоичного потока привела к термину “сигнатура” (т.е. подпись). Опираясь на принцип “временного окна” и используя импульсы пуска и останова и сигналы синхронизации от проверяемой системы, в узел логической схемы можно подать периодический набор. Этот набор подается на вход линейной последовательностной схемы, и при восприятии сигнала останова в регистре сдвига окажется “сигнатура” данного узла и конкретного тест-набора. Фактическое значение сигнатуры несущественно, но оно должно быть одним и тем же для данного узла, он стимулируется одним и тем же тест-набором и когда применяется те же самые сигналы пуска, останова и синхронизации.

 Для получения наилучших результатов при применении способа регистра сдвига требуется последовательность максимальной длины что приводит к широкому классу схем, называемых *генераторами псевдослучайной последовательности*. В 16-разрядном регистре сдвига имеется 2048 способов реализации отводов обратной связи, удовлетворяющих данному критерию. В полиноме ЦИК-16 применяется четное число входов, что приводит к группированию ошибок, а при тестировании узлов предпочтителен метод, который максимально распределяет ошибки. По этой же причине отводы не рекомендуется делать через 4 или 8 разрядов, так как они соответствуют наиболее вероятным размерам слов в микропроцессорах. Фирма Hewlett-Packard остановилась на нечетном числе входов, применив неприводимое выражение обратной связи X16 + X12 + X 9 + X7 + 1, которое соответствует характеристическому полиному X16 + X9 + X 7 + X4 + 1. Напомним, что мы хотим получить прибор широкого назначения для тестирования цифровых систем; имеются и другие характеристические выражения, которые удовлетворяют критерию, но было выбрано именно это.

 По существу, получается портативный прибор, содержащий 16-разрядный регистр сдвига с сумматором по mod 2 на входе. С помощью сигналов пуска и останова входной сигнал подается в схему, а сигнал синхронизации от проверяемой системы сдвигает данные в регистре. По окончании интервала измерения осуществляется индикация содержимого регистра как характеристической сигнатуры проверяемого узла.

 Входной двоичный поток от проверяемого узла с помощью сигналов пуска и останова подается в регистр, который синхронизируется сигналом от проверяемой системы. После прекращения двоичного потока остаток, находящийся в регистре, выводится на индикаторы в виде четырех 16-ричных символов, которые и представляют собой “сигнатуру” проверяемого узла.

 Возможна замена стандартного набора 16-ричных символов. Фирма Hewlett-Packard заменила стандартный набор 16-ричных символов, чтобы избежать путаницы между цифрой 6 и буквой *b*, и приняла следующие символы:

|  |  |  |
| --- | --- | --- |
| **ДВОИЧНЫЙ КОД** | **ИНДИКАЦИЯ** | **16-РИЧНЫЙ СИМВОЛ** |
| 0000 | 0 | 0 |
| 0001 | 1 | 1 |
| 0010 | 2 | 2 |
| 0011 | 3 | 3 |
| 0100 | 4 | 4 |
| 0101 | 5 | 5 |
| 0110 | 6 | 6 |
| 0111 | 7 | 7 |
| 1000 | 8 | 8 |
| 1001 | 9 | 9 |
| 1010 | A | A |
| 1011 | C | B |
| 1100 | F | C |
| 1101 | H | D |
| 1110 | P | E |
| 1111 | U | F |

 Большинство изготовителей сигнатурных анализаторов пользуются таким же кодированием индицируемых данных, что и фирма Hewlett-Packard.

 Такое понятие, как почти правильная сигнатура, не имеет смысла; индицируемый код 006А совершенно не связан с кодом 006С. Сигнатура может быть только правильной или не правильной.

# Какие изделия пригодны для испытания методом сигнатурного анализа.

 Как правило, в изделия, при эксплуатации которых предполагается использовать сигнатурный анализ, в процессе разработки вносят определенные средства, позволяющие производить этот анализ наиболее простыми и дешевыми приборами и повысить эффективность контроля. Прежде всего это средства, позволяющие разрывать в режиме контроля цепи обратной связи в контролируемой схеме. Когда выходит из строя один из элементов, входящих в контур с обратной связью, локализовать неисправности внутри этого контура с помощью сигнатурного анализа не удается.

 Поясним это на примере схемы, изображенной на рис. 2. Допустим, что нисправная ИМС 2. Это приводит к появлению неправильной сигнатуры на выходе схемы Т, причем на всех входах схемы сигнатуры правильные, а в точках A, B, C и D – неправильные. Указать, какой из четырех элементов схемы вышел из строя, невозможно, так как нет ни одного элемента, у которого при неправильной выходной сигнатуре были бы правильные все входные. Если снабдить схему переключателем *П1*, который в режиме тестирования размыкает цепь обратной связи, сигнатура в точке E перестает зависить от сигнатуры в точке D, и в этом случае неисправность ИМС 2 легко обнаруживается.



Рис. 2. Пример цифровой схемы с обратной связью

 На практике, как правило, не требуется разрывать контуры с обратной связью, включающие в себе всего несколько простейших логических элементов, подобные схеме, изображенной на рис. 2, так как локализация сигнатурным анализом неисправности с точностью до такого контура позволяет быстро выявить неисправность конкретного элемента другими методами. Что же касается МПС в целом, то она целиком охвачена обратной связью по контуру программного управления.

 Поэтому для применения сигнатурного анализа необходимо иметь возможность разрывать цепи обратной связи в режиме тестирования МПС. Для однокристального микропроцессора это условие реализуется отключением шины данных от входа МП. Разрыв обратной связи по шине данных можно реализовать с помощью механических переключателей или электронных ключей.

 Вторым важным условием пригодности МПС для испытаний с помощью сигнатурного анализа является наличие схем, вырабатывающих сигналы *Пуск* и *Стоп*, необходимые сигнатурному анализатору для выработки измерительного “окна”, т.е. интервала времени, в течении которого накапливается сигнатура.

 Третьим требованием является наличие в составе МПС ПЗУ, в которм содержится тестовая программа. В качестве тестовой может выступать как специально разработанная, так и определенная рабочая программа, если она, по мнению разработчика, в достаточной мере использует все устройства МПС.

 Увеличение объема аппаратуры и стоимости разработки МПС с учетом требований применимости сигнатурного анализа по сравнению с объемом и стоимостью обычной МПС не превышает в среднем 1%.

# Простой сигнатурный анализатор.

 Дешевый сигнатурный анализатор можно собрать из стандартных логических ИС. Основу его составляет 16-ти разрядный регистр сдвига с элементом исключающего ИЛИ в цепи обратной связи. Первоначально регистр переводится в нулевое состояние, которое является запрещенным состоянием для автономного ГПСП(Генератор ПсевдоСлучайной Последовательности), но из которого его можно вывести любым битом с логической 1 во входном потоке данных. Сигналы пуска, останова и синхронизации подаются из проверяемой системы вместе с сигналом от проверяемого узла. Сигнал пуска разрешает прохождение сигналов синхронизации в регистр сдвига, поэтому данные можно синхронно сдвигать в регистре. Сигналом пуска можно также клапанировать входной поток данных. Сигнал останова прекращает синхронизацию регистра сдвига и блокирует входные данные. Сигналом останова остаток из регистра сдвига передается на индикаторы.

## Структурная схема простого сигнатурного анализатора.

Структурная схема простого сигнатурного анализатора приведена на рис. 3.

 Схема иллюстрирует несколько интересных особенностей. 16-разрядный регистр сдвига реализован на двух микросхемах КР1533ИР8, а цепь обратной связи построена на двухвходовых элементах исключающего ИЛИ 1533ЛП5. Входной поток данных для улучшения формы сигналов подается на триггер Шмитта.

 В приборе широкого назначения необходимо иметь возможность устанавливать для сигналов пуска, останова и синхронизации любой активный фронт – нарастающий или спадающий. Например, в одном тесте нужен сигнал пуска с активным нарастающим фронтом, а в другом тесте он должен запускать операции спадающим фронтом. Возможность задания активного фронта сигнала обеспечивается входными элементами исключающего ИЛИ, через которые сигналы проходят в схему управления. В случае сигнала пуска переключатель S1 подсоединяется к земле или к источнику питания Vcc. Когда ключ замкнут на землю, на выходе элемента исключающего ИЛИ повторяется входной сигнал пуска. Если же S1 подключен к Vcc, сигнал на выходе представляет собой инверсию входного сигнала. Предположим, что для инициирования схемы управления всегда требуется нарастающий фронт сигнала. Тогда для удовлетворения этого требования с помощью S1 можно выбрать либо нарастающий фронт сигнала (S1 подключен к земле), либо спадающий фронт сигнала (S1 подключен к Vcc).



Рис. 3. Простой сигнатурный анализатор

 Остаток, сформированный в регистре сдвига, при восприятии сигнала останова индецируется как “сигнатура” проверяемого узла. Отметим, что информация индицируется в стандартном 16-ричном формате, а не в специальном формате фирмы Hewlett-Packard. Наличие однозначного соответствия между обоими форматами, к которому легко привыкнуть, устраняет этот недостаток.

## Секция анализа простого сигнатурного анализатора.



Рис. 4. Секция анализа простого сигнатурного анализатора

 На рис. 4. Показана секция формирования сигнатур простого сигнатурного анализатора.

 До начала работы прибора оператор вручную сбрасывает систему, при этом устанавливается начальное состояние схемы управления, а 16-разрядный регистр сдвига переводится в нулевое состояние. После сброса сигнал пуска разрешает прохождение сигнала синхронизации в регистр сдвига, который синхронно сдвигает данные до появления сигнала останова. Светодиод в схеме управления показывает, что прибор формирует сигнатуру. Во время процесса формирования сигнатуры индикаторы сигнатуры находят ся в погашеном состоянии.

Линия входных данных подается на триггер Шмитта для предотвращения ложного срабатывания схемы.

Схема управления выполнена на двух RS - триггерах, трех 2-х входовых элементах И-НЕ и двух инверторах. Сигналом сброса схема управления запрещает прохождения сигнала синхронизации на сдвиговые регистры и подготавливается к приему сигнала пуска. После прихода сигнала пуска, схема управления начинает пропускать сигналы синхронизации на сдвиговые регистры и подготавливается к приему сигнала останова. После прихода сигнала останова, снова блокируется прохождение сигналов синхронизации на сдвиговый регистр, при этом на индикаторах отображается информация полученная в сдвиговом регистре.

## Схема индикации сигнатуры зафиксированной в анализаторе.

 Схема индикации сигнатуры, зарегистрированной в анализаторе, приведена на рис. 5. Секция индикатора и анализатора работают асинхронно. Выходы регистра сдвига после завершения формирования сигнатуры подаются на дешифраторы. Выходы каждого дешифратора подаются на соответствующие элементы индикации (семисегментный индикатор).

 

 Рис. 5. Секция индикации простого сигнатурного анализатора

 Данные индицируются в стандартном 16-ричном формате, причем цифра 6 имеет “хвостик”, позволяющий отличить ее от малой буквы *b.*

 Описанный анализатор относительно прост, но является вполне работоспособным прибором и стоит примерно в 10 раз дешевле анализаторов, выпускаемых промышленностью.

## Описание элементной базы используемой для создания простого сигнатурного анализатора.

 Для реализации схемы приведенной на рис. 4, 5 использовались интегральные микросхемы ТТЛ серии КР1533, так же микросхемы серии КР514 и полупроводниковые идикаторы.

 Далее приводится перечень используемых микросхем и их обозначения на схемах.

|  |  |  |
| --- | --- | --- |
| ***Название микросхемы*** | ***Функциональное назначение*** | ***Обозначения на схемах*** |
| КР1533ТЛ2 | Шесть триггеров Шмитта-инверторов | DD1 |
| КР1533ЛП5 | Четыре двухвходовых логических элемента ИСКЛЮЧАЮЩЕЕ ИЛИ | DD2, DD8 |
| КР1533ЛН1 | Шесть логических элементов НЕ | DD3 |
| КР1533ТР2 | Четыре RS-триггера | DD4 |
| КР1533ЛА3 | Четыре логических элемента 2И-НЕ | DD5 |
| КР1533ИР8 | Восьмиразрядный сдвиговый регистр с последовательной загрузкой и араллель-ной выгрузкой | DD6, DD7 |
| КР514ИД4 | Дешифратор семисегментного индикатора с отображением 16 различных знаков | DD9, DD10, DD11, DD12 |
| АЛС304А | Полупроводниковый семисегментный индикатор с общим катодом | HG1, HG2, HG3, HG4 |

# Тестирование в режиме свободного счета с использованием сигнатурного анализатора.

## Системное ядро. Режим свободного счета.

Для поддержания работоспособности системы не должно быть отказов в некоторых ее компонентах. Эти важнейшие компоненты в совокупности называются системным ядром, и в него обычно входят ЦП, системный генератор синхронизации, шина управления и шина адреса.

 Цифровой компьютер можно рассматривать как ядро, окруженное периферийными схемами, и ядро должно работать, чтобы можно было проверить остальные компоненты системы. К счастью, системное ядро проверяется довольно просто, если в системе предусмотрены средства, которые позволяют разорвать шину данных и ввести команду в ЦП. Обычно в ЦП вводится какая-нибудь разновидность *холостой команды*, например NOP или MOV A,A. После этого системное ядро можно перевести в режим СВОБОДНОГО СЧЕТА.

 ЦП осуществляет операцию считывания из памяти для выборки следующей команды. Она всегда интерпретируется как команда NOP, что заставляет процессор перейти к следующему адресу памяти и произвести еще одну операцию считывания. ЦП вынужден считывать команду “нет операции” из каждой ячейки памяти, в результате чего на шине адреса формируются все возможные двоичные коды. При просмотре сигналов в каждой линии шины адреса можно установить факт ее отказа, проявляющийся в замыкании на землю или на питание, в обрыве линии или в замыкании на другую линию шины адреса.

 Если по линиям шины адреса передаются правильные сигналы, можно считать, что системное ядро функционирует. Отказ в системном генераторе синхронизации или неисправная линия шины управления почти наверняка не дадут правильной работы в режиме свободного счета. Тест свободного счета обеспечивает простой метод тестирования некоторых важных компонентов в микропроцессорной системе и применим к любому микропроцессору. Режим свободного счета имеет большое значение для сигнатурного анализа.

## Использование сигнатурного анализатора в режиме свободного счета.

 Сигнатурный анализатор можно использовать для проверки ядра системы, образуя сигнатуры узлов в режиме свободного счета.

 При проведении любого теста с применением сигнатурного анализатора необходимо решить, какие сигналы от проверяемой системы следует использовать в качестве сигналов пуска, останова и синхронизации. В промышленных сигнатурных анализаторах зонд для касания узла имеет логический пробник, который дает визуальную индикацию активности. Конечно, индикатор пробника не дает возможности определить природу действий в узле, но он показывает наличие или отсутствие сигналов в проверяемом узле.

 Предположим, что исследуется система с 8-битным микропроцессором, имеющим 16-битную шину адреса. В цикле свободного счета на шине адреса возникают все двоичные наборы, которые циклически повторяются. Благодаря периодичности такой тип активности идеально подходит для сигнатурного анализа. На старшей линии A15 шины адреса действует низкий уровень для одной половины всех адресов и высокий для другой половины. Следовательно, между соседними нарастающими фронтами сигнала на линии A15 находится один полный цикл шины адреса. Сигнал с этой линии можно использовать как сигналы пуска и останова анализатора. Первый из них осуществляет запуск, а второй - останов. Остаток, образованный в регистре сдвига между этими событиями, подается на индикатор в качестве сигнатуры проверяемого узла. В режиме свободного счета все команды осуществляют считывание из памяти, и сигналы для анализатора можно взять с линии READ. Анализатор настраивается на нарастающие фронты входов пуска, останова и синхронизации, поэтому данные синхронно проходят через регистр сдвига по заднему фронту сигнала READ в течении одного полного цикла шины адреса. Необходимы подключения показаны на рис. 6.



Рис. 6. Получение сигнатур в режиме свободного счета

 До взятия сигнатур от узлов в системе сам сигнатурный анализатор и подключения входных сигналов контролируются по сигнатурам земли и питания Vcc. Регистр сдвига а анализаторе инициализируется на нуль до регистрации любых данных. Когда пробник касается земли, вход данных всегда находится в состоянии логического нуля 0, которое не изменяет начального состояния регистра сдвига. По окончании цикла регистрации данных остаток в регистре сдвига будет нулевым. Это состояние может изменить только входной сигнал логической 1, которого, очевидно, не может быть при контроле сигнатуры земли. Следовательно, земля всегда дает сигнатуру 0000, которую можно считать ее “характеристической сигнатурой”. Однако положительное питание Vcc всегда воспринимается как состояние логической 1, которое изменяет начальное состояние регистра сдвига. Остаток, образующийся по окончании регистрации данных, зависит от числа состояний синхронизации между сигналами пуска и останова и будет различным при изменении запускающих сигналов. При конкретном подключении сигналов пуска, останова и синхронизации сигнатура Vcc будет одной и той же, поэтому ее называют “характеристической сигнатурой” для данного подключения входов. Но, разумеется, она будет получаться различной при других подключениях управляющих входов и (или) выборе других активных фронтов. Когда от проверяемого узла получается такая же сигнатура, как и от Vcc, может оказаться что из-за отказа он закорочен на шину питания Vcc. Однако иногда и от исправных узлов получается такая же сигнатура, как и характеристическая сигнатура Vcc. Проще всего различать эти две ситуации по индикатору логического пробника – в исправном узле он вспыхивает, показывая наличие сигналов в узле. Если же индикатор не вспыхивает, следует предположить наличие отказа.

*Сигнатуры адреса в режиме Шина данных Шина адреса*

 *свободного счета D0 X A0 UUUU A8 HC89*

 *D1 X A1 5555 A9 2H70*

*Переключатели Подключе- D2 X A2 CCCC A10 HPP0*

*сигнатурного ния D3 X A3 7F7F A11 1293*

*анализатора D4 X A4 5H21 A12 HAP7*

 *D5 X A5 0AFA A13 3C96*

*Пуск  A15 D6 X A6 UPFH A14 3827*

*Останов  A15 D7 X A7 52F8 A15 755P*

*Cинхро-*

*низация*  *READ*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Сигнатура Vcc равна 0001*



Рис. 7. Документирование сигнатур

 Для любой системы, рассчитанной на сигнатурный анализ, должен существовать документ, в котором приведены сигнатуры всех узлов. В режиме свободного счета, хотя он и сканирует все адресное пространство, будут разрешены не все микросхемы, так как микропроцессор выполняет только операции считывания из памяти. Например, входной порт не разрешается, и получить от него значимые сигнатуры невозможно. Шина данных отключена от процессора и также не дает значимых сигнатур (за исключением некоторых специальных условий). Набор сигнатур берется от заведомо исправной системы и документируется. В начале списка сигнатур показываются подключения входов пуска, останова и синхронизации, а также их активные фронты (нарастающий или спадающий). Кроме того, здесь же даются характеристические сигнатуры для указанных подключенных входов. Эта информация необходима для настройки анализатора и контроля подключений по сигнатурам, полученным от Vcc и земли. После этого берутся сигнатуры от узлов, и результаты сравниваются с приведенными в документе сигнатурами. Пример таблицы сигнатур для режима свободного счета приведен на рис. 7.

Таблица представляет собой только небольшую часть полного перечня сигнатур, которые берутся для каждой микросхемы в системе. В начале таблицы показаны используемые от системы сигналы.

 В данном примере характеристическая сигнатура Vcc равна 0001; ее необходимо проверить до взятия от системы остальных сигнатур. В режиме свободного счета сигналы на шине данных бессмысленны, что показано в таблице в виде набора состояний “безразлично” (X). Однако шина адреса проверяется, поэтому приведены сигнатуры для всех линий шины адреса.

 Далее в документе находятся диаграммы разводки выводов всех микросхем, и у каждого вывода показана его сигнатура. Земля всегда имеет характеристическую сигнатуру 0000, которая приводится как GND. Чтобы показать, что сигнатура 0000 допустима для вывода и “отличается” от сигнатуры земли, после сигнатуры находится буква B. Она показывает, что светодиод, находящийся в зонде логического пробника, при взятии сигнатуры будет вспыхивать. Примером служит сигнатура у вывода 18 микросхемы IC2. В режиме свободного счета сигнатуры не многих вывода ИС не имеют смысла и показываются на диаграммах в виде X (см. пример у вывода 3 IC4). Еще одна часто встречающаяся ситуация отражена у вывода 1 IC2. Здесь сигнатура равна 0000, но светодиод в зонде пробника не вспыхивает. Вывод 1 в данном тесте всегда имеет уровень логического 0, который дает такую же сигнатуру, как и земля; указание 0000 на диаграмме подчеркивает, что вывод не закорочен на землю. Если вывод закорочен на землю, следует указывать GND.

 Поиск неисправности с применением сигнатурного анализа заключается в определении возможной области с неисправностью и проверки сигнатур до обнаружения неправильной сигнатуры. Пользуясь принципиальной схемой и таблицами сигнатур, неисправность прослеживают “назад” до получения правильной сигнатуры. Неисправность существует в той части схемы, которая находится между последней неправильной и первой правильной сигнатурами.

# Тестирование ПЗУ в режиме свободного счета.

 Хотя главное назначение свободного счета заключается в проверке системного ядра, он применим и для контроля ПЗУ. В режиме свободного счета на шине адреса переодически формируется все двоичные наборы. Подаваемая в ЦП холостая команда заставляет его выполнять операции считывания по каждому адресу. ПЗУ содержит только фиксированные команды, которые в режиме свободного счета последовательно выдаются на шину данных. Используя линию разрешения кристалла ПЗУ для сигналов пуска и останова, а управляющую линию READ для сигнала синхронизации анализатора, можно проверить содержимое любого системного ПЗУ.



Рис. 8. Тестирование ПЗУ в режиме свободного счета

Анализатор регистрирует только данные, относящиеся к проверяемому ПЗУ, хотя процессор сканирует все адресное пространство. На рис. 8 показано, каким образом в режиме свободного счета проверяется одна из системных ПЗУ с применением сигнатурного анализатора.

 Аналогичный тест для микросхем ОЗУ применять нельзя, так как их содержимое не фиксировано, и для проверки работы ОЗУ разработаны другие тесты. Поскольку ЦП выполняет только операции считывания из памяти, невозможно проверить каналы ВВ, особенно в том случае, если ВВ отображен на адресное пространство памяти, в режиме свободного счета проверить можно, так как ЦП считает обращение к ним операциями считывания из памяти. При этом необходимо управлять входами в порты, для чего обычно применяется тестовый прибор с возможностью задания известных двоичных наборов. Для проверки выходных портов потребуется операция записи, которой в режиме свободного счета нет.

# Тест–циклы сигнатурного анализа.

 Для проверки тех частей системы, которые недоступны режиму свободного счета, необходимо написать и выполнить специальные программы. Каждая из них предназначена для проверки одной части системы, например входного порта, и обычно состоит всего из нескольких строк ассемблерного кода. Такие тест-программы сигнатурного анализа обычно помещаются в ПЗУ, которое находится в системе, но при нормальной работе не используется. Тест-ПЗУ приводится в действие либо переключением линии OE от первого системного ПЗУ с последующим сбросом системы, либо вводом в процессор команды RESTART и размещением тест-ПЗУ по адресу рестарта. Набор тест-программ обычно организован как цикл, который переодически выполняется при включении тест-ПЗУ в работу. В начале полного тест-цикла предусматривается формирование импульса на линии сигнала пуска сигнатурного анализатора, а в конце тест-цикла формируется импульс на линии сигнала останова анализатора. Часто для обоих сигналов пуска и останова используется одна и та же линия, что устраняет необходимость формирования отдельного сигнала останова. В микропроцессорах типа 8080/Z80 в качестве сигнала пуска-останова обычно применяется старшая линия А15 шины адреса, а само формирование сигнала осуществляется фиктивной командой ввода или вывода. В этих микропроцессорах адреса портов ВВ формируются только в младьшей половине шины адреса, но адрес дублируется и на старшей половине шины. Поэтому считывание или запись в порт ВВ с адресом 8016 вызовет появление импульса на линии А15.

|  |  |  |
| --- | --- | --- |
| TESTLOOP: | DI | ;Запретить прерывание |
|  | IN 80H | ;Выдать импульс при считывании |
|  | OUT 80H | ;Выдать импульс при записи |

 Приведенную простую последовательность команд (в мнемониках микропроцессора 8080) можно использовать для инициирования набора тест программ сигнатурного анализа. Команды IN и OUT вызывают появление импульса на линии A15 сначала в операции считывания, а затем в операции записи. В зависимости от выполняемого теста вход синхронизации в сигнатурный анализатор подается с линии READ или WRITE шины управления.

 Простой тест входного порта реализуется следующим образом:

|  |  |  |  |
| --- | --- | --- | --- |
| OUTTEST: | XRA |  | ;сбросить регистр A  |
|  | STC | A | ;установить в 1 бит переноса |
| LOOP: | RAL |  | ;сдвинуть еденичный бит влево |
|  | OUT | (04) | ;выдать двоичный набор в выходной порт |
|  | JNC | LOOP | ;повторить если тест не закончен |

 Принцип теста заключается в сдвиге состояния логической “1” по всем восьми линиям ВВ выходного порта. Таким образом, до перехода к следующему тесту будет произведено 8 операций записи в выходной порт. При использовании линии A15 для сигналов пуска и останова, а линии WRITE для входа синхронизации берутся сигнатуры с каждой выходной линии и сравниваются с задокументированными значениями. Аналогичный тест можно написать для проверки каждой входной линии входного порта, но вначале необходимо задать их известные состояния. Обычно для этого к входным линиям подключается тест-прибор, позволяющий устанавливать состояния входных линий.

# Встроенные средства для сигнатурного анализа

 В идеальном случае тестирование вычислительной системы с помощью сигнатурного анализа необходимо предусматривать еще на этапе проектирования. При разработке системы следует встроить средства реализации свободного счета и выбора тест-циклов­ из ПЗУ. Когда управляющие сигналы для анализатора должны формироваться из комбинации нескольких внутренних сигналов, следует ввести в систему необходимые схемы. После окончания разработки можно получить и задокументировать полные таблицы сигнатур для различных тестов.

 Многие системы разработаны без всяких встроенных средств тестирования, но их можно приспособить для сигнатурного анализа. Прежде всего необходимо ввести схемы, реализующие режим свободного счета системного ядра, чтобы проконтролировать важнейшие компоненты компьютера или микропроцессорной системы. К счастью, в большинстве систем микропроцессор находится в панельки и его можно вынуть. Панелька микропроцессора имеет все сигналы и напряжения питания, которые требуются для реализации свободного счета. В простейшем случае нужно вынуть микропроцессор и вставить в его панельку переходник, в котором встроены разрыв шины данных и подача в микропроцессор холостой команды. После этого с другой стороны переходника вставляется микропроцессор. Для конкретного микропроцессора в отделе контроля нужно разработать переходник для встраивания в любую систему, в которой применяется данный микропроцессор. При эксплуатации нескольких систем с различными микропроцессорами потребуется изготовить соответствующие переходники для каждого микропроцессора. Таким образом, все возможности режима свободного счета реализуются в любой системе независимо от того, проектировалась ли она с учетом сигнатурного анализа или без учета его.

 Тесты свободного счета проверяют только часть системы, а для стимулирования других ее частей необходимо написать и разместить в ПЗУ тест-программы. Обычный подход заключается в том, чтобы заменить первое системное ПЗУ на тест-ПЗУ. После сигнала системного сброса микропроцессор производит обращение к этому ПЗУ, и тест-программы проверяют всю систему.

 Тест-программы сигнатурного анализа выполняются как последовательность коротких фрагментов, следующих друг за другом. По окончании последовательности процессор переходит к началу, образуя полный тест-цикл. Для локализации отказов в системе при проверке отдельных компонентов обычно требуются свои сигналы пуска и останова. Большинство этих сигналов уже имеется в системе; например, каждое ПЗУ можно проверить подключив к управляющим линиям пуска и останова конкретный сигнал выбора кристалла от дешифратора адреса. ОЗУ обычно проверяется путем выполнения первоначальной последовательности операций записи, в результате чего во всех ячейках ОЗУ будет храниться известный двоичный набор. Операции записи контролируют линии шины адреса, подключенные к ОЗУ, и линии шины управления, разрешающие работу микросхемы. Убедившись в функционировании этих линий, можно выполнить последовательность операций чтения, которые контролируют линии подключения шины данных и проверяемого ОЗУ. Любые сигналы используемые в качестве сигналов пуска и остнова сигнатурного анализатора, должны опираться на результаты предыдущих тестов, контролирующих систему по частям. Режим свободного счета проверяет системное ядро, а работоспособное ядро обеспечивает подлежащие проверке сигналы. В большинстве случаев успешный контроль свободного счета свидетельствует о правильном функционировании самого микропроцессора.

# Ограничения сигнатурного анализатора

 Сигнатурный анализатор регистрирует только те события, которые синхронны с сигналом синхронизации, используемым для получения сигнатур от узлов системы. Микропроцессор является синхронным конечным автоматом, и при сигнатурном анализе проверяется большинство его частей. Однако некоторые события происходят асинхронно с системной синхронизацией, и их нельзя проконтролировать с помощью сигнатурного анализа. Последовательная линия связи в стандарте RS232C обычно работает асинхронно с системной синхронизацией, что препятствует применению сигнатурного анализа. В этом случае на вход синхронизации анализатора можно подать сигнал синхронизации универсального асинхронного приемопередатчика и получить сигнатуры последовательной линии связи. Однако компьютер необходимо запрограммировать на выдачу в линию периодического сигнала, чтобы в тесте участвовала известная периодическая двоичная последовательность.

 Компьютер в шинной структурой можно считать замкнутой системой с обратной связью, в которой по шине адреса осуществляется запрос данных, а реакция выводится с шины данных. Проблема в такой структуре заключается в том, что отказ распространяется по всей петле, вызывая получение кажущихся плохих сигнатур в тех местах, где отказ отсутствует. Сигнатурный анализ не показывает, в каком месте временного окна впервые возникает отказ, что можно было бы использовать для локализации неисправного компонента. Плохая сигнатура по истечении временного окна не показывает “историю” прослеживания отказа. Поэтому, когда много микросхем осуществляют вывод на общую шину и появляется неправильная сигнатура, сигнатурный анализ не может указать, в какой именно микросхеме возник отказ.

 Усовершенствованной формой сигнатурного анализа явился трассовый анализ, который локализует отказ в окне пуска-останова. Сигнатурный трассовый анализатор программируется (обычно вручную) на последовательность промежуточных сигнатур и их моменты появления во временном окне. При работе прибора первое несовпадение измеренных данных с хранимой информацией заставляет сигнатурный трассовый анализатор остановитьс

я и индицировать момент появления ошибки. После этого отказа можно подазревать в той микросхеме, которая работала в момент его появления. Правильные данные для сигнатурного трассового анализа, как и сигнатуры, необходимо получить от заведомо исправной системы и задокументировать их. При этом сигнатуры разделяются на короткие секции с соответствующим увеличением объема документируемой информации. Однако дополнительная документация потребуется только для тех частей системы, которые невозможно адекватно проверить, пользуясь обычным сигнатурным анализом.

# Заключение

 Сигнатурный анализ представляет собой простой способ тестирования, заключающийся в проверке отдельных узлов схемы и сравнения их сигнатур с документированными значениями. Умелое его применение позволяет быстро идентифицировать неисправность в системе, а затем локализовать неисправный компонент.

 Для сигнатурного анализа очень важна хорошая документация. Любые модификации в системе влекут за собой необходимость повторного получения всех сигнатур в системе, поэтому при ее модификациях и усовершенствованиях необходимо составлять скорректированные таблицы сигнатур. Важность поддержания своевременно обнавляемой документации невозможно переоценить. Часто системы модифицируются, а изменения в принципиальные схемы не вносятся. Это обстоятельство само по себе создает проблемы тестирования, но если не корректируются и таблицы сигнатур, сам способ оказывается неработоспособным и бесполезным, так как полученные сигнатуры не имеют смысла без соответствующих правильных документированых значений.

# Список используемой и рекомендуемой литературы:

1. Уильямс Г.Б. Отладка микропроцессорных систем: Пер. с англ. – М.: Энергоатомиздат, 1988.
2. Микропроцессоры: системы программирования и отладки / В.А.Мясников, М.Б.Игнатьев, А.А.Кочкин, Ю.Е.Шейнин; Под ред. В.А.Мясникова, М.Б.Игнатьева. – М.: Энергоатомиздат, 1985.
3. Фергусон Дж., Макари Л., Уилльямз П. Обслуживание микропроцессорных систем: Пер. с англ.– М.: Мир, 1989.
4. Кирьянов К.Г., Соловейчик Э.Б. К проэктированию РЭА, ориентированной на диагностику сигнатурным анализом. – Техника средств связи. Сер. Радиоизмерительная техника, 1980, вып. 1 (26), с 9 - 84.
5. Пароль Н.В., Кайдалов С.А. Знакосинтезирующие индикаторы и их применение: Справочник. – М.: Радио и связь, 1989.
6. И.И.Петровский, А.В.Прибыльский, А.А.Троян, В.С.Чувелев. Логические ИС КР1533, 1534. Справочник. В двух частях. Часть1, 2.
7. Пухальский Г.И., Новосельцева Т.Я. Цифровые устройства: Учебное пособие для втузов.­– СПб.: Политехника, 1996.