ФОРМ
научно-производственная фирма

121108, г. Москва, ул. Ивана Франко, 4
Тел/факс (499)144 79 44, (499)146 11 75, (495)642 07 54
info@form.ru, http://www.form.ru

Рекомендации
по реализации режима «Match»
на тестере FORMULA 2K
ФРМИ 2.653.020 И901

Редакция 1

1 Назначение режима «Match»

Режим **Match** (более правильно называемый «Pattern Matching» или **«**поиск состояния») является разновидностью (особым способом выполнения) функционального контроля ФК на тестере **FORMULA 2K**.

Режим **Match** обычно используется для синхронизации ФК (тестера) с объектом контроля. При этом осуществляется **поиск необходимого состояния** объекта контроля путём сравнения текущего его состояния с заданным эталоном.

При выполнении ФК в «нормальном» (не **Match**) режиме браком (ошибкой) считается несовпадение **ожидаемых** (заданных в тестовой последовательности ТП) и **реальных** значений откликов тестируемой микросхемы.

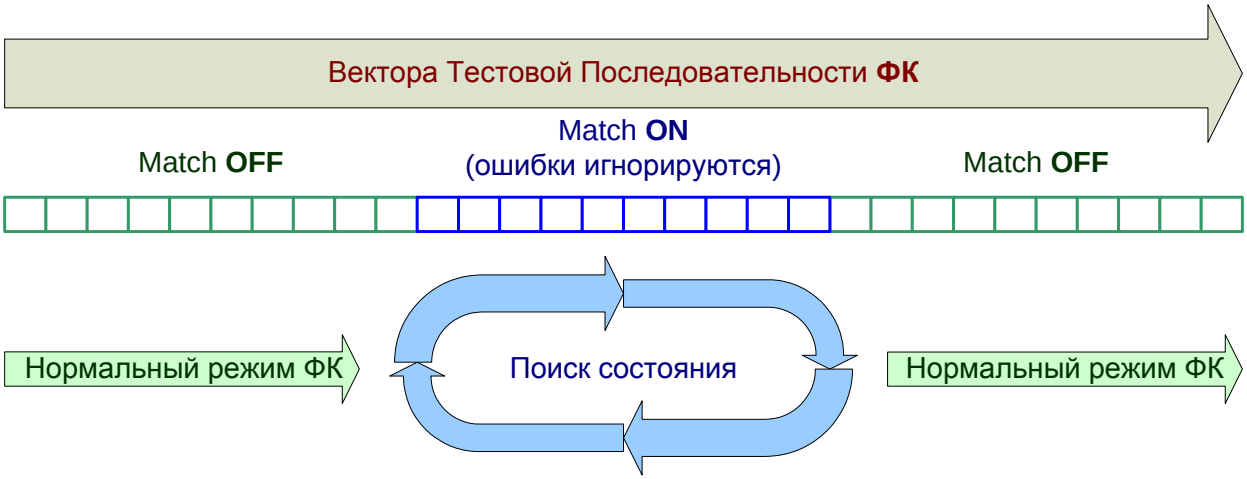
В режиме же **Match** такое несовпадение всего лишь сигнализирует о том, что ожидаемое состояние микросхемы пока не достигнуто. В этом смысле оно противоположно событию «брак» и не может считаться ошибкой ФК. Другими словами, в режиме **Match** интересующее событие происходит, наоборот, при **совпадении** текущего состояния микросхемы с ожидаемым.

Примеры объектов контроля, при функциональном контроле которых бывает необходимо применять режим поиска состояния (**Match**):

- счетчики (делители), не имеющие сигнала сброса. Перед тестированием необходимо «прогнать» счетчик до определенного (обычно нулевого) состояния;
- микропроцессоры, в процессе своего включения выполняющие операцию аппаратной инициализации внутренних узлов с недокументированным временем выполнения. Перед тестированием необходимо «дождаться» перехода микросхемы в рабочее состояние, признаком которого обычно является появление определенной комбинации на её выходах;
- микросхемы последовательных интерфейсов с «кадровой» («фреймовой») организацией обмена. Для синхронизации с трактом передачи объекта контроля необходимо «выявить» начало кадра в последовательном потоке данных. Признаком такого начала обычно является выдача «синхрослова» (уникальной последовательности нулей/единиц определенной длины).

2 Общие принципы организации режима Match

При поиске состояния (**Match**) выполнение некоторого участка ФК зацикливается до тех пор, пока на выходах испытуемой микросхемы не возникнет необходимая комбинация сигналов. После этого ФК выходит из цикла и дальнейший тест выполняется обычным (стандартным) образом.



В теле цикла **Match** должны вырабатываться необходимые воздействующие на микросхему сигналы (например, формирование синхроимпульса), переводящие её в следующее состояние Т.о. по крайней мере один вектор цикла должен содержать символ(ы) воздействия **0/1/Q/J**.

Аналогично, хотя бы один вектор цикла **Match** должен осуществлять **контроль** ожидаемой (эталонной) комбинации на выходах микросхемы, т.е. содержать символ(ы) **L/H/Z**.

Зацикливание ФК обычно осуществляется с помощью команды условного перехода и продолжается до тех пор, пока не сбросится флаг ошибки на векторе с контролем. Это означает, что нужное состояние микросхемы найдено.

ВНИМАНИЕ: в данном случае активация флага ошибки не является браком в общепринятом смысле этого слова, а лишь сигнализирует о том, что ожидаемое состояние пока не достигнуто. Поэтому в режиме поиска возникшие «браки» должны аппаратно игнорироваться (не приводить к останову ФК по ошибке). Для этого служат специальные команды входа в режим **Match** и выхода из него. Они активизируются по мере необходимости в процессе выполнения ФК.

3 Особенности реализации режима Match на тестере FORMULA 2K

Вхождение в режим **Match** в процессе выполнения ФК на тестере **FORMULA 2K** осуществляется с помощью команды «установка режима работы» **Mode** (мнемоника **Md**) с аргументом **DontStopOnHWErr.** При этом **выключается** аппаратный флаг остановки **ФК** в случае возникновения ошибки. Другими словами, в режиме **Match** ошибки игнорируются.

Выход из режима **Match** осуществляется также с помощью команды «установка режима работы» **Mode**, но с аргументом **StopOnHWEr**. При этом аппаратный флаг остановки при возникновении ошибки **включается**. Это соответствует нормальному режиму работы ФК, при котором ошибки фиксируются (если противоположное не указано напрямую, например, аргументом `StopError=0` стандартной процедуры `start_beg_hand`).

Зацикливание режима **Match** удобнее всего осуществлять с помощью команды «условного перехода по аппаратной ошибке» **IfNotHWErr** (мнемоника **IfN**). Переход в этой команде осуществляется по следующим правилам:

- если ошибки (брака) **нет**, то осуществляется переход на указанную в поле «Метка/Адрес» редактора ФК метку, или на адрес вектора, вычисленный по заданному в этом поле смещению (которое может быть как положительным, так и отрицательным числом). Рекомендуется в поле команды указывать метку, а не смещение;
- если ошибка **есть**, линейно выполняются следующие за данной командой **IfN** вектора (т.е. перехода нет).

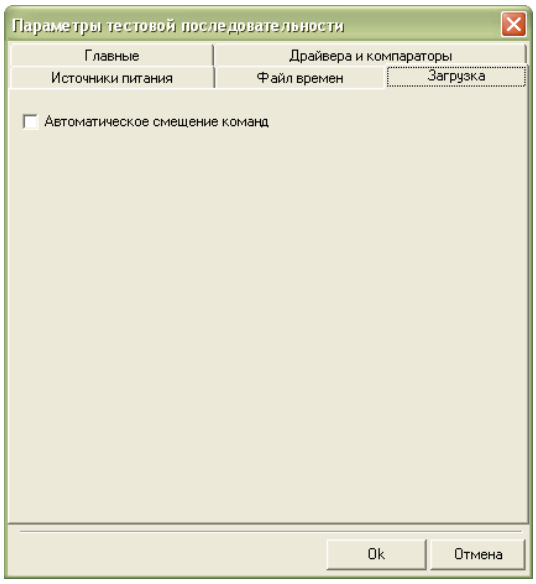
При необходимости (например, для организации режима **Match** с более сложными критериями поиска состояния) возможно использовать также команду условного перехода по регистру R7 (мнемоника **IfR7**), которая обычно применяется «в паре» с командой работы с флагами **LogOp** (мнемоника **LOp**).

При организации режима **Match** на тестере **FORMULA 2K** следует учитывать, что аппаратная часть ФК тестера имеет конвейерную организацию. Это означает, что контроль откликов и выполнение команд может занимать несколько тактов (векторов) ФК. Поэтому при формировании тестовой последовательности следует учитывать следующие несложные правила:

- команда **IfNotHWErr** при осуществлении перехода анализирует аппаратную ошибку, возникшую за 4-ре вектора до неё;

- команда **IfR7** не учитывает модификации регистров, произведенные в двух векторах перед ней;
- после команд **IfNotHWErr** и **IfR7** осуществляется «выбег» ФК на 2 вектора, и только после этого осуществляется непосредственно переход. Для нормальной работы эти два вектора должны быть «холостыми», т.е. не содержать контроля и команд.
- условный переход невозможен через границу блока памяти в 4096 векторов (через вектор с адресом, кратным 0x1000);

ПРИМЕЧАНИЕ: указанные выше замечания верны для отключенного режима «автоматическое смещение команд» редактора ФК. Использовать автоматическое смещение при организации режима Match **не рекомендуется**.

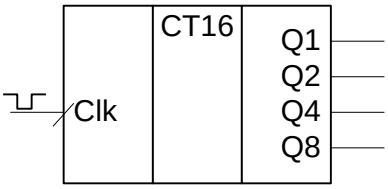


Более подробно об особенностях работы команд ФК см. документ «Редактор программ функционального контроля ФРМИ 2.653.020 Д22».

4 Пример реализации режима Match для 4-х разрядного счетчика без сброса

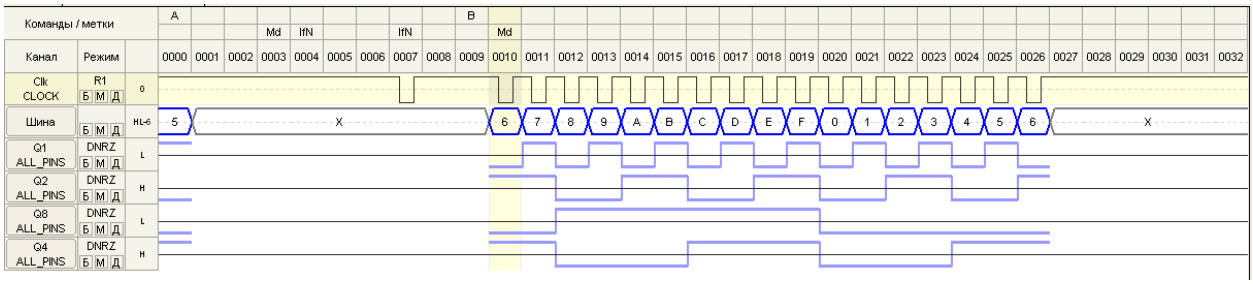
В качестве примера рассмотрим реализацию режима **Match** для простого абстрактного объекта контроля — 4-х разрядного счетчика, не имеющего сигнал «Сброс» (Reset).

Ниже приведено условное обозначение такого счетчика:



При включении питания счетчик обычно устанавливается в непредсказуемое (случайное) состояние. До выполнения полноценного тестирования его необходимо «предустановить» в необходимое состояние путем т.н. «прогона», т.е. подачи на счетный вход **Clk** некоторого количества импульсов. Для определенности предположим, что для нормального тестирования счетчика его исходное состояние (комбинация сигналов на выходах **Q**[8,4,2,1]) должно быть равным 5-ти (**0101**).

Тестовая последовательность, реализующая функциональный контроль данного счетчика, представлена на следующей диаграмме (в виде окна редактора ФК).



Для удобства восприятия эта же тестовая последовательность представлена в текстовом формате «НЕВОД-F»:

```
//        CQQQQ
//        14812
//        k
%A
$0000000 1HLHL,
         1XXXX,
         1XXXX,
         1XXXX #Mode DontStopOnHWErr,
         1XXXX #IfNotHWErr $000009,
         1XXXX,
         1XXXX,
         0XXXX #IfNotHWErr $000000,
         1XXXX,
%B
         1XXXX,
         0HLLH #Mode StopOnHWErr,
         0HLHH,
         0LHLL,
         0LHHL,
         0LHLH,
         0LHHH,
         0HHLL,
```

```
0HHHL,
0HHLH,
0HHHH,
0LLLL,
0LLHL,
0LLLH,
0LLHH,
0HLLL,
0HLHL,
0HLLH,
1XXXX,
1XXXX,
1XXXX,
1XXXX,
1XXXX,
1XXXX,
```

Вектора с 10-го по 26-й тестовой последовательности предназначены непосредственно для тестирования счетчика.

Предполагается, что к началу тестирования счетчик уже находится в исходном состоянии «5».

Тестирование осуществляется путем многократной подачи счетного импульса, тем самым осуществляется перебор всех возможных состояний счетчика, которые последовательно увеличиваются на единицу. Контролируемые результаты сравниваются с ожидаемыми.

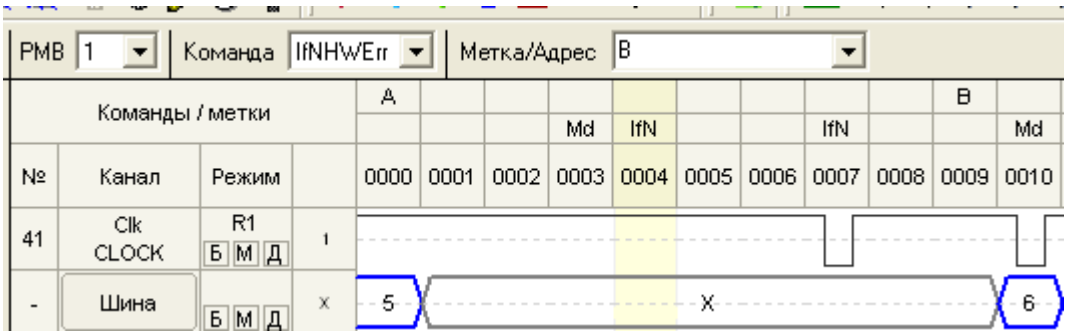
Вектора с 0-го по 9-й предназначены для организации поиска необходимого исходного состояния «5», т.е. непосредственно для организации режима **Match**.

В векторе 0, который помечен меткой «A», осуществляется контроль выходных сигналов счетчика.

ПРИМЕЧАНИЕ: Состояние контролируемых сигналов **Q**[8,4,2,1], записанных в этом векторе ТП, должно соответствовать тому, что должно быть у микросхемы после выполнения поиска. В нашем случае оно равно 5-ти, т.е. **LHLH**.

Если при выполнении ФК текущее состояние счетчика в этом векторе не совпадёт с ожидаемым, равным **LHLH**, то активируется флаг ошибки.

Пройдя через конвейер длиной 4-ре вектора, ошибка будет проанализирована в 4-м векторе с помощью команды условного перехода **IfN**. Адрес перехода команды указывает на метку «B», которой помечен вектор 9.



После выполнения команды **IfN** ФК совершает «выбег» еще на 2 вектора, поэтому непосредственное ветвление осуществляется на векторе 6.

Если ошибки в процессе ФК нет, то это означает, что уже достигнуто необходимое состояние счетчика, равное **LHLH**. В этом случае с 6-го вектора ФК «перепрыгнет» на 9-й вектор, тем самым осуществив выход из цикла **Match**.

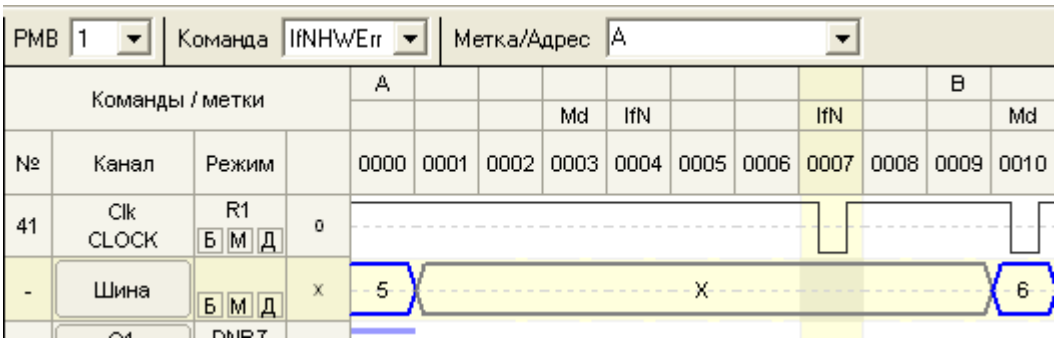
В противном случае (ошибка есть, состояние еще не достигнуто) прыжка не будет и после 6-го вектора будет выполнен следующий, 7-й вектор.

В этом векторе тестируемый счетчик переводится в следующее состояние путем подачи единичного отрицательно импульса на счетный вход **Clk**.

Также необходимо возвратиться на начало цикла **Match** (вектор 0) для повторной проверки состояния микросхемы на предмет совпадения с эталоном.

Для возвращения на начало цикла команда безусловного перехода **GoTo** в общем случае не совсем подходит (переходит только на адреса, кратные 0x1000).

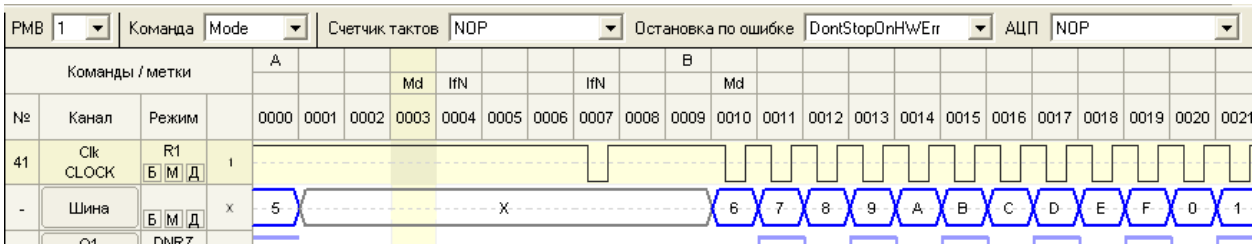
Тем не менее, такой переход вполне удобно осуществить с помощью всё той же команды **IfN.** Для этого в векторе, расположенном за 4-ре до вектора 7 (т.е. в векторе 3), не следует ставить контроль (**X**). Тогда к моменту исполнения команды **IfN** флаг ошибки будет гарантировано равным нулю (нет контроля — нет ошибки). В этом случае команда **условного** перехода «вырождается» в команду **безусловного** перехода, но без ограничений по адресу перехода, свойственных команде **GoTo**.



Поле «Метка/Адрес» вектора 7 указывает на метку «A». Это означает, что после вектора 7 будет осуществлен «выбег» ещё на 2 вектора, и с вектора 9

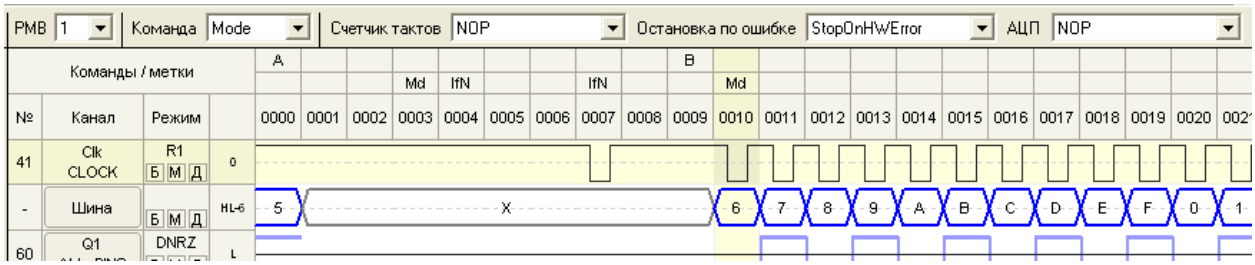
выполнен (всегда!) «прыжок» на вектор 0. Т.е. на начало цикла **Match**, что и требовалось.

Команда **Mode** в векторе 3 имеет параметр «Остановка по ошибке» равный **DontStopOnHWErr**, и тем самым осуществляет вхождение в режим **Match.** После первого её исполнения, на время поиска состояния, будет происходить игнорирование аппаратных ошибок.



ПРИМЕЧАНИЕ: допускается многократное (в цикле) выполнение команды **Mode** с одним и тем же параметром.

Напротив, команда **Mode** в векторе 10 имеет параметр «Остановка по ошибке» равный **StopOnHWErr**, и тем самым осуществляет возвращение в нормальный режим выполнения ФК после выхода из цикла **Match**.



ПРИМЕЧАНИЕ: тестер не поддерживает отдельную команду условного перехода «по ошибке» (а только «по её отсутствию» — **IfN**), которая теоретически позволяла бы организовывать цикл **Match** с помощью одной команды перехода. Поэтому, в общем случае, для организации цикла необходимо использовать две команды переходов (**IfN**), что, впрочем, ни в коей мере не влияет на работоспособность режима **Match**.

Т.о. при выполнении рассмотренной тестовой последовательности с 0-го вектора независимо от того, в каком состоянии «оказался» счетчик после включения питания, после выполнения режима **Match** он гарантированно перейдет в состояние **0101**. Тем самым обеспечивается успешное выполнение основного теста счетчика (с 10-го вектора) с результатом «годен».

ВНИМАНИЕ: следует понимать, что при запуске измерительной программы существует потенциальная опасность «зависания» ФК («невыхода» из цикла **Match**) в случае, например, неисправности объекта контроля (он не реагирует на воздействующие сигналы и никогда не перейдет в искомое состояние).

В этом случае рекомендуется принудительно останавливать ФК по тайм-ауту.

Для этого, например, после запуска ФК (`start_beg_hand`) можно выполнить задержку времени (`DelayMs`) с величиной, гарантирующей выход из режима **Match** при самых неблагоприятных условиях (максимальном количестве повторений цикла). После чего проверить, завершен ли ФК (`check_stop`). Если не завершён (ФК «завис»), выполнить принудительную остановку ФК (`stopPP`).

ПРИМЕЧАНИЕ: существует возможность принудительного выхода из цикла **Match** и непосредственно в процессе выполнения ФК, без внешнего сброса по тайм-ауту. Для этого удобно использовать команды **IfR7** и **LOp**, с помощью которых можно организовать, например, подсчет количества выполненных циклов «на лету». Более подробное рассмотрение этого способа выходит за рамки данного документа.

5 Заключение

Программно-аппаратные ресурсы, предоставляемые тестером **FORMULA 2**K, позволяют организовать полноценный режим **Match** для микросхем, требующих выполнения поиска состояния в процессе функционального контроля.