Метод граничного сканирования - Boundary-Scan

# (стандарт IEEE 1149.1)

С середины 70-х годов структурное тестирование печатных плат основывалось на физическом доступе к устройствам и межсоединениям на плате с помощью так называемого «ложа из гвоздей» (рис. 1). С развитием технологии изготовления печатных плат, стали уменьшаться расстояния между соседними медными дорожками – стало более рискованно каждый раз класть плату на гвозди. Появились двухсторонние платы, и инженерам пришлось прикладывать гвозди с двух сторон. Вопрос о физическом доступе был исчерпан с появлением многослойных печатных плат.

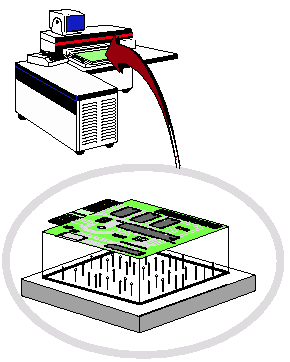


Рисунок 1

С такими проблемами столкнулись в середине 80-х большинство производителей. Для поиска возможных решений была создана группа JTAG (Joint Test Action Group), которая с помощью 200 ведущих фирм в течение 5 лет разработала стандарт Boundary-Scan IEEE 1149.1 (IEEE – Institute of Electrical and Electronic Engineers).

В основу принципа была положена концепция размещения последовательного сдвигового регистра по границам устройства (рис. 2) Причем, сами ячейки такого регистра располагаются непосредственно между первичными входами/выходами устройства и логическим ядром.

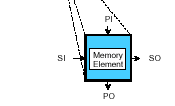
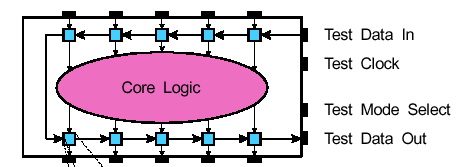


Рисунок 2

Таким образом, в устройство добавляется 4 контакта: TDI (Test Data In – вход тестовых данных), TDO (Test Data Out – выход тестовых данных), TMS (Test Mode Select – выбор тестового режима) и TCK (Test Clock – тестовая синхронизация).

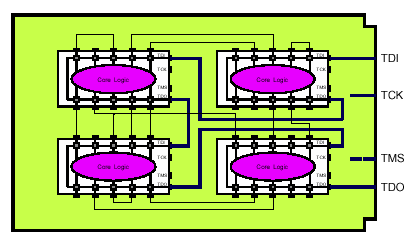


Рисунок 3.

Сущность механизма сканирования сходна с методом Scan Path. Однако, есть существенные различия. Во-первых, логика ядра может быть любой, в том числе содержать последовательностные структуры (не нужно разбивать устройство на комбинационную часть и память). Во-вторых, ячейки сканирования располагаются как до, так и после логики.

Несколько устройств на плате могут быть объединены в единый последовательный путь сканирования (рис. 3). Это можно сделать путем соединения TDO с TDI следующего устройства. Таким образом, на плате появляется также 4 дополнительных вывода: TDI и TDO, и общие для всех устройств TCK и TMS.

Таким образом, технология граничного сканирования при относительно небольших аппаратурных затратах позволила реализовать нечто вроде «виртуальных гвоздей», что обеспечивает не только 100% наблюдаемость, но и 100% управляемость входов/выходов устройств.

Несомненно, технология Boundary-Scan является значительным прорывом вперед в DFT. Рассмотрим архитектуру устройства, которую описывает стандарт IEEE 1149.1.

Общая схема устройства представлена на рис. 4.

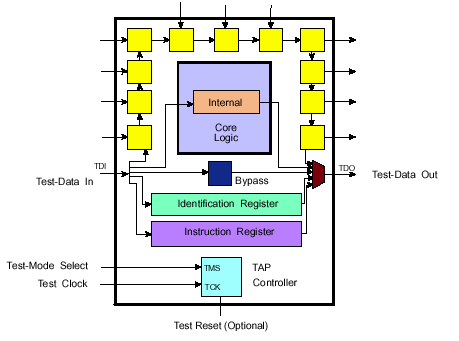


Рисунок 4.

Само устройство, реализующее требуемые функции (которое, как мы видим, может содержать и некие внутренние регистры), помещается в обрамление из тестовой логики, которое составляют:

* Набор из 4 обязательных тестовых ножек: TDI, TDO, TMS и TCK. Также возможно добавление еще одной необязательной тестовой ножки – TR (Test Reset – сброс тестовой логики). Все вместе они называются TAP (Test Access Port – тестовый порт доступа).
* Ячейки граничного сканирования на каждой линии первичного входа или выхода, соединенные между собой, таким образом, чтобы организовать последовательный сдвиговый регистр.
* Однобитный регистр обхода (Bypass register).
* Необязательный 32-битный идентификационный регистр, с возможностью хранения в нем неизменяемого кода устройства.
* Регистр команд (Instruction Register), в котором хранится текущая команда. Разрядность регистра команд должна быть больше или равна 2.
* TAP controller. По сути, это конечный управляющий автомат для тестовой логики.

Как видно из схемы, тестовая логика предоставляет возможность выбирать путь прохождения данных от TDI к TDO: либо через регистр команд (Instruction Register), либо через регистр данных (Data Register). Регистром данных в каждый момент времени может быть один из следующих регистров:

* Последовательный сдвиговый регистр граничного сканирования (Boundary-Scan register).
* Какой-либо внутренний регистр, предусмотренный разработчиком устройства (Internal register).
* Регистр обхода (Bypass register).
* Идентификационный регистр (Identification register).

Рассмотрим регистр сканирования. Ячейки граничного сканирования (Boundary-Scan cells) подразделяются на два типа: входные (размещаются на тех первичных входах устройства, где временной фактор критичен – например, на входе синхронизации) и обычные. Схема входной ячейки показана на рис. 5.

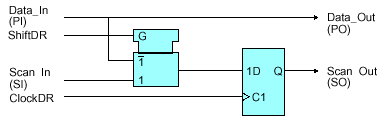


Рисунок 5.

Как видно из этой схемы, сигнал проходит по линии данных без задержек. Однако, там где это возможно на входных линиях, а также на всех выходных линиях располагают обычные ячейки. Структура типичной ячейки граничного сканирования представлена на рис. 6.

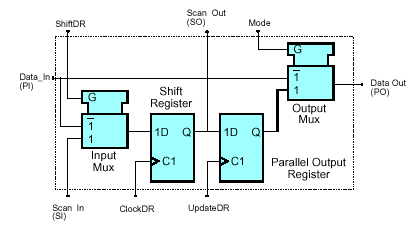


Рисунок 6

Рассматриваемая ячейка состоит из двух D-триггеров, работающих по прямому фронту, двух мультиплексоров “2-в-1”, четырех входов и двух выходов. Назначение структурных элементов схемы:

* Data\_In и Data\_Out – вход и выход (относительно ячейки) линии данных, на которую эта ячейка и помещена. (PI – parallel input, PO – parallel output).
* Scan In и Scan Out – соответственно, вход и выход сканируемых данных (SI – serial scan input, SO – serial scan output). Т.е. на SI данные приходят с SO предыдущей ячейки (либо с TDI, если это первая ячейка в пути), проходят через первый D-триггер и выходят на SO, чтобы попасть на SI следующей ячейки (или на TDO, если это последняя ячейка в пути).
* ShiftDR – определяет источник информации для первого D-триггера: Data\_In (режим Capture) либо Scan In (режим Shift).
* Mode – определяет, что пропускать на линию данных Data\_Out: сигнал с Data\_In (режим Normal) либо сигнал со второго D-триггера (режим Update).
* ClockDR – синхронизирующий вход первого D-триггера, т.е. для режимов Capture и Shift.
* UpdateDR – синхронизирующий вход второго D-триггера (режим Update)

Ниже приведена таблица режимов функционирования ячейки:

|  |  |  |  |
| --- | --- | --- | --- |
| Режим | ShiftDR | Mode | Функционирование |
| NORMAL | - | 0 | PI → PO |
| CAPTURE | 0 | - | PI → SO по фронту на ClockDR |
| SHIFT | 1 | - | SI → SO по фронту на ClockDR |
| UPDATE | - | 1 | SO → PO по фронту на UpdateDR |

Как видно из структуры ячейки, путь сканирования TDI → SI1 → SO1 → …→ SOi-1 → SIi → SOi → SIi+1 → … → TDO физически не пересекается с линиями данных PI→PO ячеек граничного сканирования. Таким образом, регистр граничного сканирования является “прозрачным” для самого устройства. Для осуществления операций сканирования отсутствует необходимость переключаться в специальный тестовый режим. Влияние тестовой логики на функционирование устройства сведено к минимуму. На входных линиях задержка прохождения сигнала может отсутствовать вообще, а на выходных – равна задержке выходного мультиплексора (Output Mux).

Внутренний регистр (Internal register), предусмотренный разработчиком устройства, например, для хранения результирующей сигнатуры проверки устройства, если в нем реализованы средства встроенного самотестирования.

Регистр обхода (Bypass register) – простой, но очень важный элемент в рассматриваемой архитектуре. Он создает кратчайший путь между TDI и TDO, не затрагивая при этом все остальные регистры. Необходимость такого регистра обусловлена ранее оговоренной возможностью соединения нескольких устройств в последовательный путь сканирования. Если нам необходимо иметь доступ не ко всем, а только к некоторым устройствам в цепочке, то остальные устройства устанавливаются в режим Bypass, образуя минимальные задержки прохождения сканируемых данных. Bypass register состоит из одного мультиплексора “2 в 1” и одного D-триггера, синхронизированного прямым фронтом (рис. 7). Отсутствует параллельный вывод данных (в связи с чем, сигнал Update\_DR отсутствует в схеме), но есть некоторая особенность с операцией Capture – в регистр может записываться аппаратно реализованное значение 0 (т.е. производится захват значения 0).

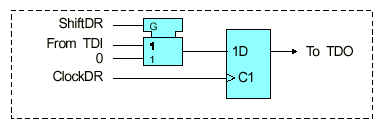


Рисунок 7

Регистр Идентификации (Identification register) – необязательный 32-битый регистр с режимами сбора данных и сдвига (рис. 8). 32 бита идентифицируют устройство через следующие поля:

* Бит 0 (младший разряд) всегда равен 1.
* Биты 1 – 11 идентифицируют фирму-изготовителя устройства. Используется краткая форма кода идентификации JEDEC.
* Биты 12 – 27 обеспечивают 16-битное поле свободного формата с серийным номером устройства.
* Биты 28 – 31 обеспечивают 4-битное поле свободного формата, чтобы описать до 16 различных версий того же основного устройства.

Регистр команд (Instruction Register) состоит из собственно сдвигового регистра, некоторой декодирующей логики (в зависимости от количества и типов реализуемых команд) и секции хранения декодированной команды, в которую как только команда поступает, она сразу же поступает на выполнение (рис. 9). Стандарт IEEE 1149.1, во-первых, предписывает, что есть только 3 обязательные команды, необходимые для функционирования аппарата Boundary-Scan, а все остальные являются необязательными (дополнительными), во-вторых, обязательные и дополнительные команды в стандарте только описываются функционально, а их реализация на аппаратном уровне оставлена полностью на усмотрение разработчика. Последнее, кстати, относится ко всему тестовому обрамлению устройства.

Следующие три команды должны обязательно быть реализованы тестовой логикой :

* BYPASS (обход) – команда, устанавливающая Bypass register между TDI и TDO. Код команды – все единицы (число разрядов регистра команд ≥2, поэтому для регистра минимальной длины код команды – 11). Режимы Shift и Capture с «0».

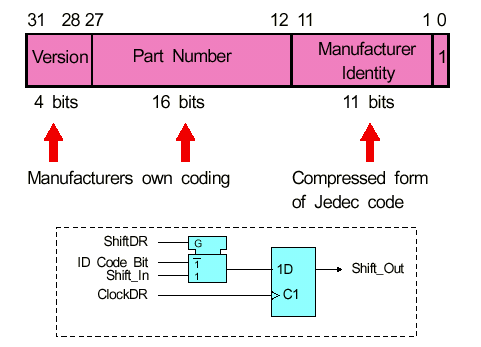


Рисунок 8

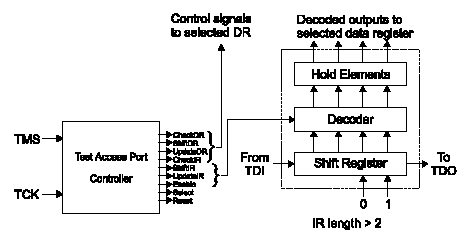


Рисунок 9.

* EXTEST (внешнее тестирование) – команда, выбирающая сдвиговый регистр и предназначенная для тестирования внешних соединений между устройствами на плате. Режимы Update (для ячеек, находящихся на первичных выходах устройства), Capture и Shift. Код команды – все нули (00).

Выполнение команды предполагает отключение ядра кристалла от внешних контактов (Mode=1).

* SAMPLE/PRELOAD (последовательность/предварительная загрузка) используется для сканирования значений сигналов, захваченных с внешних контактов устройств в штатном режиме (параллельно выполнению обычных функций устройств), а также для загрузки в регистр сканирования информации, используемой в дальнейшем для тестирования внешних соединений или других функций. Код команды не фиксирован. Режимы Capture и Shift.

Выполнение команды не предполагает отключение ядра кристалла от внешних контактов (Mode=0).

TAP контроллер (TAP controller) (рис. 10 и 11) – необычный автомат. Он имеет вход синхронизации TCK, вход данных TMS и 16 внутренних состояний. Также он имеет 9 выходов (3 – для управления регистром команд, 4 -для управления регистром данных и 2 – общего назначения). Причем переход в некоторое состояние активизируется последовательностью двоичных сигналов на входе TMS.

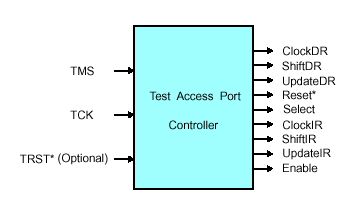


Рисунок 10 - TAP контроллер

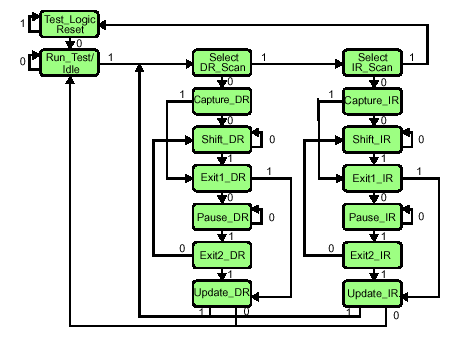


Рисунок 11 – двоичный граф переходов TAP контроллера

Рассмотрим пример использования архитектуры Boundary-Scan для тестирования межсоединений. Пусть дана схема из 3-х устройств с BS-архитектурой (рис. 12), необходимо протестировать целостность линий между 2-м и 3-м устройствами. Последовательность действий состоит из 3-х этапов:

1. Устанавливаем Instruction Register (IR), между TDI и TDO каждого устройства. Это достигается с помощью подачи определенной последовательности на вход TMS TAP контроллеров.
2. Предполагая, что размер IR = 2 бит, вводим последовательность “110000”. Это означает, что в IR 1-го устройства будет загружена команда BYPASS (“11”), а в IR 2-го и 3-го – команда EXTEST (“00”).
3. Подаем очередную последовательность на TMS, которая инициирует выполнение загруженных команд. Таким образом, в 1-м устройстве между TDI и TDO будет включен регистр Bypass, а в двух других - Boundary-Scan register. После этого устройства готовы к выполнению команды EXTEST. Фазы выполнения команды EXTEST:

Shift – задвигается входная тестовая последовательность на Chip2 (SI → SO).

Update – задвинутые в ячейки значения фиксируются на выходах Chip2 (SO → PO).

Capture – отклик на тестовую последовательность записывается на Chip3 PI(Chip3) = PO(Chip2), PI(Chip3) → SO(Chip3).

Shift – сдвигается отклик с Chip3 (SI → SO) и одновременно задвигается новая тестовая последовательность на Chip2 и т.д..

Хотя EXTEST позволяет производить тестирование без привлечения каких-либо дополнительных команд, чаще всего она используется совместно с SAMPLE/PRELOAD, выполняющей роль команды, загружающей информацию в регистр сканирования (в режимах Shift (SI → SO) или Capture (PI → SO)).

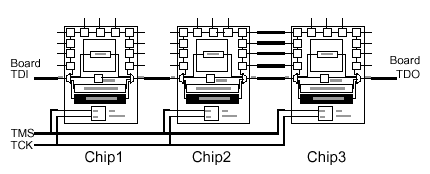


Рисунок 12

Следует заметить, что прежде чем тестировать систему, необходимо проверить исправность тестера, т.е. целостность регистра сдвига и остальных регистров данных.

Необходимо понимать, что Boundary-Scan является новым подходом к тестированию систем в том смысле, что с развитием технологии надежность устройств стала гораздо выше, чем надежность соединений между ними. Поэтому основной задачей такой архитектуры, согласно стандарту IEEE 1149.1 является тестирование межсоединений (проверка наличия, размещения и связи соединенных компонентов, а также обнаружение обрывов соединений и замыканий между устройствами).

Преимущества технологии Boundary-Scan:

* при современном уровне интеграции ИС, затраты на реализацию метода в среднем составляют около 2% площади кристалла;
* обеспечивает простоту управления конфигурированием сложных систем, включающих дочерние платы, многокристальные модули и т.п.
* позволяет организовывать on-line тестирование;
* обеспечивает доступ к регистрам, шинам, выводам;
* простой доступ к BIST возможностям устройств;
* облегчает тестирование не приспособленных к сканированию устройств, памяти.

Стандарт IEEE 1149.1 принят в 1990 году, а его усовершенствованная версия IEEE 1149.1а – в 1993. В стандарт IEEE 1149.1 входит BSDL – Boundary Scan Description Language – язык описания организации и архитектуры граничного сканирования в рамках одиночной БИС. Он построен как подмножество языка VHDL. Одной из основных задач, возлагаемых на язык, является созданием программных средств, автоматизирующих процесс создания тестов. HSDL – Hierarchical Scan Description Language (разработка фирмы Texas Instrument) – язык, ориентированный на описание потоков в группе БИС, объединенных JTAG интерфейсом. HSDL является расширением языка BSDL, полностью с ним совместим, входит в состав стандарта IEEE 1149.1а и построен как подмножество языка VHDL.