

Методы отладки и тестирования интегральных вычислительных систем

Кафедра ВТ, Университет ИТМО 2014

Кустарев П.В.

Общая информация

Рабочий учебный план М.2.2:

Лекции – 17 ч. (9 занятий)

Лабораторные работы – 34 ч. (17 занятий = 3 ЛР)

ЭКЗАМЕН

Информация:

<http://isu.ifmo.ru>

<http://embedded.ifmo.ru/forum/> \Дисциплины\Методы
отладки и тестирования ИВС\ 6113, 6114, 6115 Осень 2014

Литература:

Организация разработки интегральных и встраиваемых вычислительных систем

ВсС = встраиваемые электронные ИУС

ИВС = цифровые микросхемы: ИМС(IC, ASIC),
СнК(SoC, NoC)

Процесс разработки ВcC и ИМС/СнК (IC,ASIC/SoC).

1. Разработка Технического задания (формирование потребностей (customer/user needs), спецификация требований (requirements))
2. Проектирование (design) ВcC и ИМС/СнК.
3. Отладка(debug), валидация/ратификация (validation) и верификация (verification).
4. Реализация (implementation)
5. Тестирование (testing)

Организация проекта ИМС

- Выбор/разработка процедуры проектирования и производства ИМС, обеспечивающих реализацию требований и спецификации в условиях доступных технологий и средств проектирования и производства.

Валидация, верификация и тестирование

ЭТО НЕ ОДНО И ТО ЖЕ:

- ВАЛИДАЦИЯ проверяет результат проектирования
- ВЕРИФИКАЦИЯ проверяет процесс проектирования
- ТЕСТИРОВАНИЕ проверяет конечный продукт

**НО МОГУТ ИСПОЛЬЗОВАТЬСЯ ОДИНАКОВЫЕ
МЕТОДЫ И СРЕДСТВА**

Валидация

Валидация (design validation) – проверка на соответствие потребностям (needs) и вытекающим из них «внешним» требованиям (requirements): система должна работать так, как от нее ожидают.

Выделяют:

- Валидацию работы ПО;
- Валидацию работы аппаратуры;
- Валидацию работы системы в комплексе.

Верификация

Предиктивный (предсказательный) анализ, удостоверяющий, что разработанный продукт (например, ИМС/СнК), если будет произведен, соответствует проектным требованиям (requirements) и спецификации.

- Проверяет корректность и качество процесса проектирования;
- Реализуется методами: формальными, симуляции или эмуляции.

Выполняется до производства

Тестирование (Testing)

Это этап производственного процесса, который должен подтвердить, что произведенный продукт (например, ИМС/СнК) соответствует результатам проектирования (технической документации) и не имеет производственных дефектов.

- Проверяет корректность производственного процесса;
- Тестируется каждое изделие;
- Делится на 2 стадии: генерация тестов (на этапе проектирования) и применение тестов – тестирование (на этапе производства)

Отладка (Debug)

Отладка – выявление и устранение ошибок в проектируемой системе (устройстве и/или программном обеспечении) на этапе проектирования.

Отладка аппаратуры может выполняться:

- на «виртуальной» модели (до производства);
- на физическом прототипе (FPGA или испытательный образец ИМС).

Методы

ЭТАП	МЕТОДЫ
ВАЛИДАЦИЯ	Симуляционное моделирование, проверка формальными методами, виртуальное или физическое прототипирование
ВЕРИФИКАЦИЯ	
ОТЛАДКА	
ТЕСТИРОВАНИЕ	Физическое прототипирование, стендовое тестирование, натурное тестирование, встроенное тестирование и т.п.

Методы валидации и верификации

- Валидация/верификация функциональных (IP-) ядер (core-level);
- Верификация интерфейсов;
- Временная верификация (timing verification);
- Косимуляция аппаратуры и ПО;
- Эмуляция аппаратуры;
- Аппаратное прототипирование.

Проектирование СнК (SoC)

- Объединяет проектирование цифровых и аналоговых аппаратных блоков и ПО
 - Базируется на использовании ядер (Core-Based Design) (ЦП, DSP, Memory, Logic-Core)
 - Имеет жесткие ограничения в части реального времени, энергопотребления, технологической сложности кристаллов и другие.
 - Высокая сложность и стоимость физического (аппаратного) прототипирования
-

Основной вес проектирования SoC переносится на этап абстрактного «системного» проектирования с упором на валидацию и верификацию компьютерных моделей.

Системная верификация/валидация СНК

Функциональное моделирование SoC	Функциональное моделирование SoC	Instruction-Accurate модели ЦПУ	C, C++, PC-native
		Bus Functional Model	TLM/SystemC
		Logic Behavior Model	SystemC/SystemVerilog/HDL
		Validation/Verification Environment	SystemC
	Временное моделирование	Cycle-Accurate модели ЦПУ	C, C++, PC-native
		Bus Timing Model	TLM/SystemC/HDL
		Logic Behavior Model	SystemC/SystemVerilog/HDL
		Validation/Verification Environment	SystemC

Отладка и тестирование методом граничного сканирования

ИВС = цифровые микросхемы:
ИМС(IC, ASIC), СнК(SoC, NoC)

Граничное (периферийное) сканирование

Вид структурного и функционального тестирования плат и микросхем, основанный на прямом управлении портами микросхем , а также специальными встроенными инструментальными блоками микросхем через специальный инструментальный интерфейс.

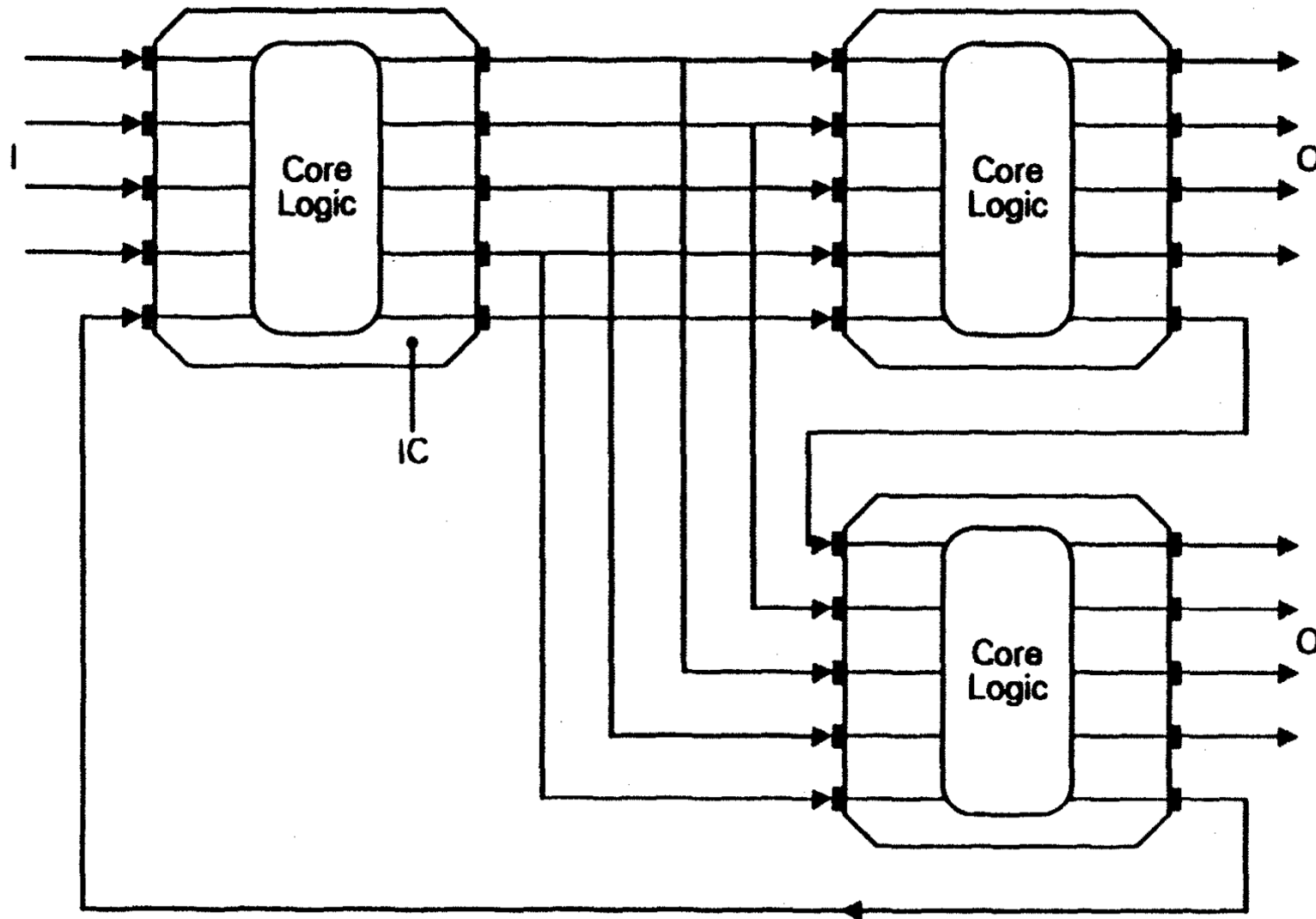
Требует встраивания в микросхемы специальных блоков граничного сканирования.

РЕАЛИЗАЦИЯ:

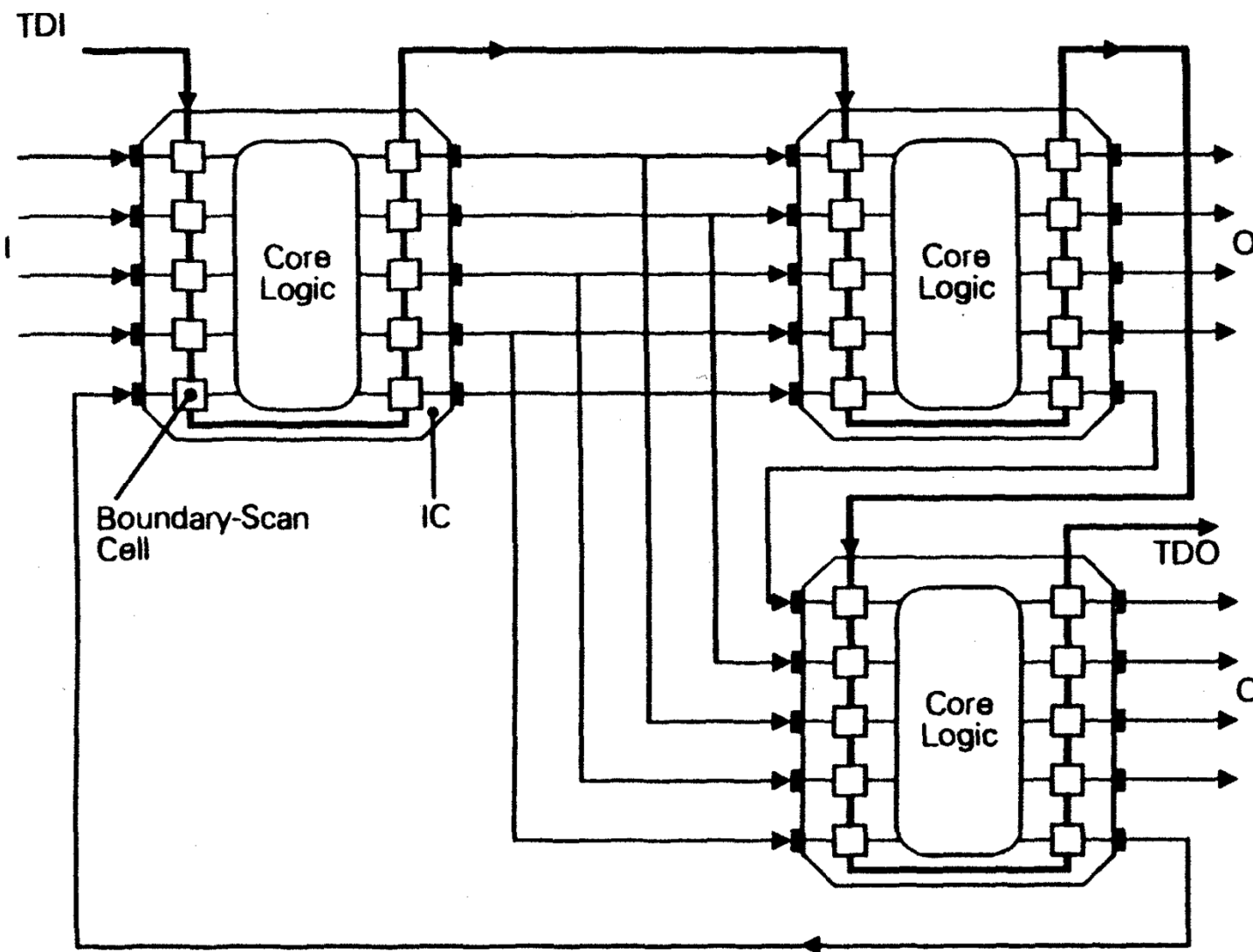
1149.1-2013 (JTAG = Joint Test Action Group) - IEEE Standard for Test Access Port and Boundary-Scan Architecture

(<http://standards.ieee.org/findstds/standard/1149.1-2013.html>)

Задача тестирования соединений на плате и микросхем



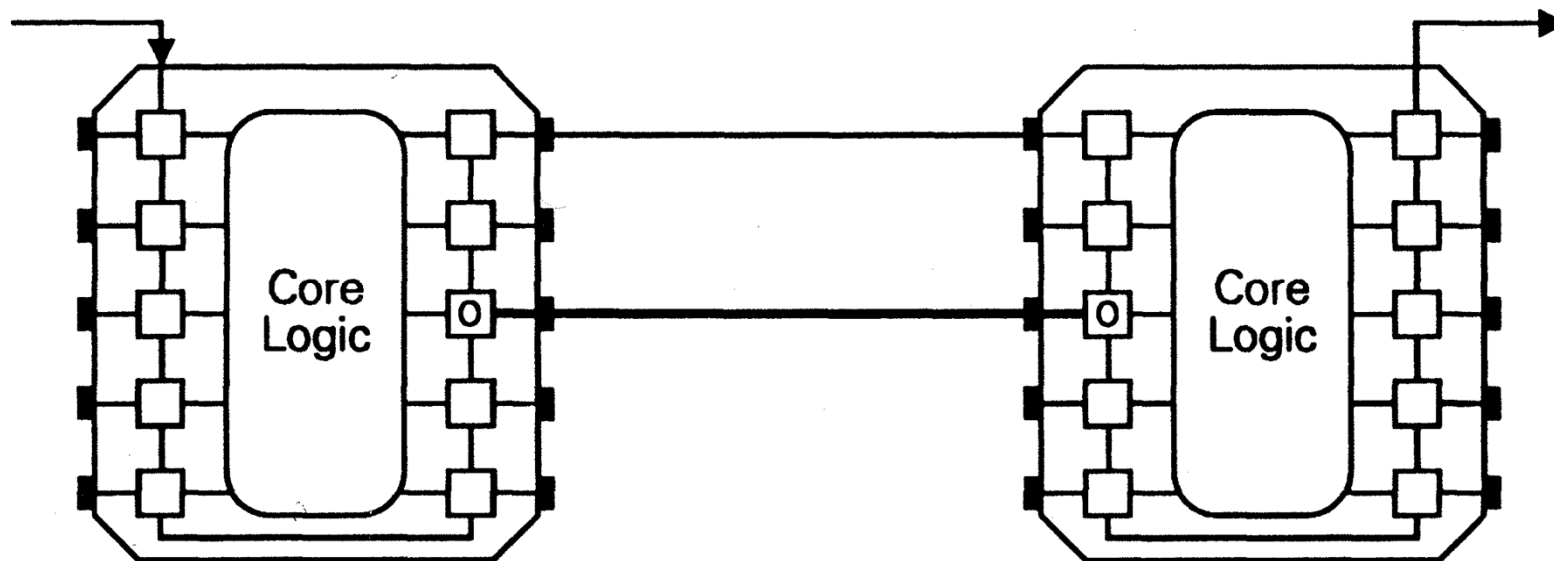
Принцип граничного тестирования (Boundary-Scan Testing - BST)



BS-тестирование соединений

Test Data Input (TDI)

Test Data Output (TDO)



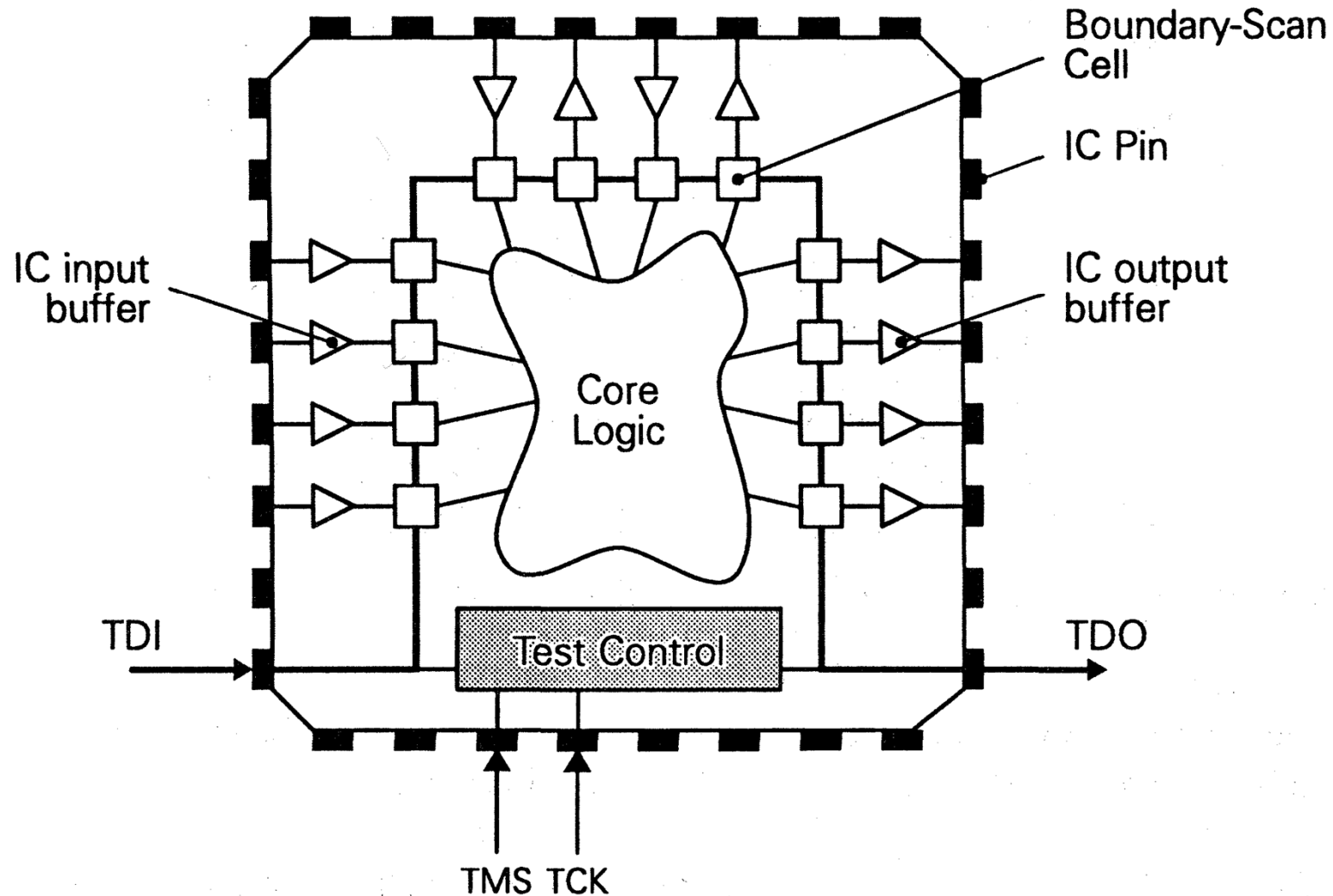
Шаг 1. *Shift-DR*: сдвиг данных через TDI до ячейки, подключенной к выходному порту

Шаг 2. *Update-DR*: запись значения бита из ячейки сдвигового регистра в выходной порт

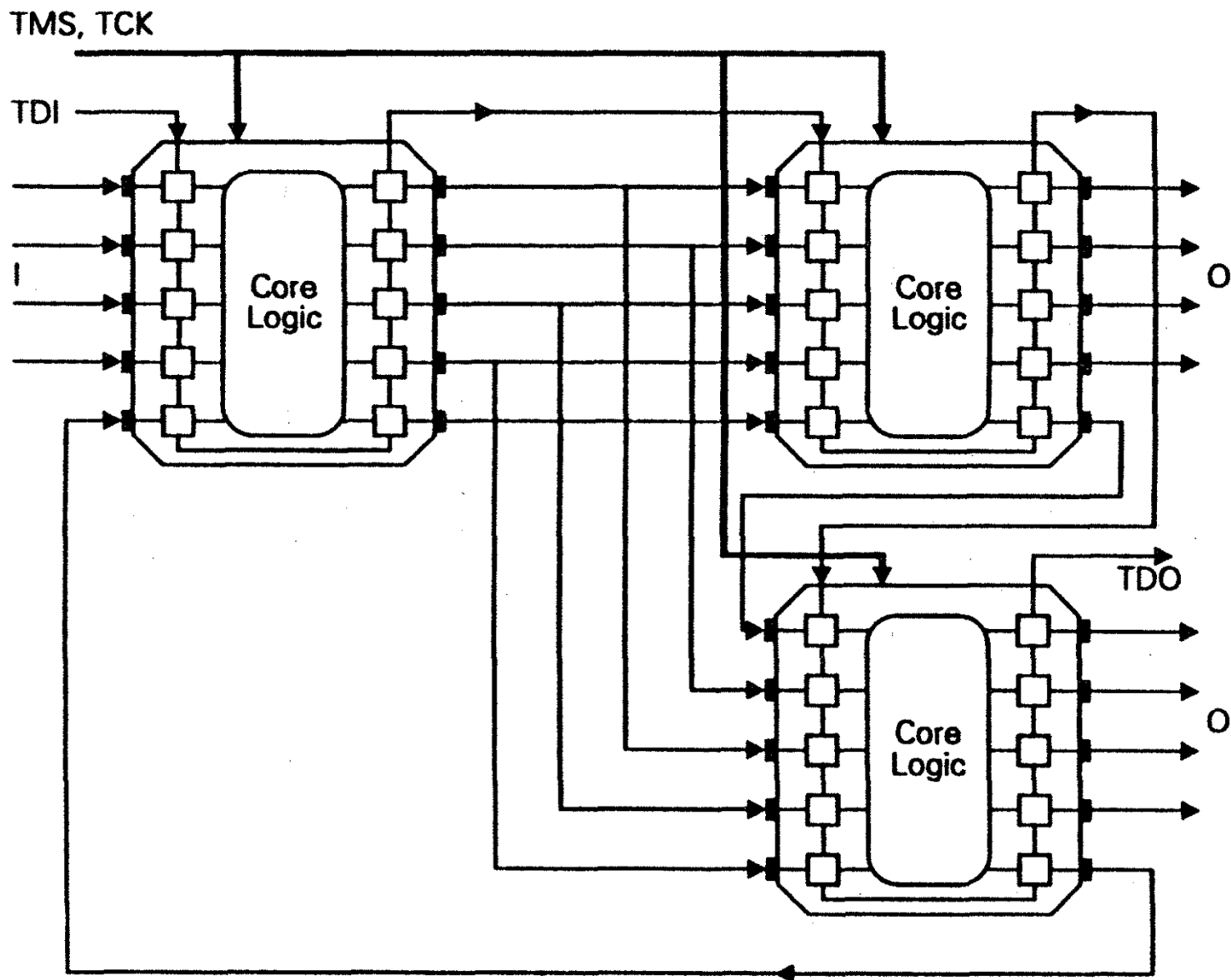
Шаг 3. *Capture-DR*: чтение значения из входного порта в ячейку сдвигового регистра.

Action 4. *Shift-DR*: сдвиг считанного значения из ячейки сдвигового регистра через TDO для анализа.

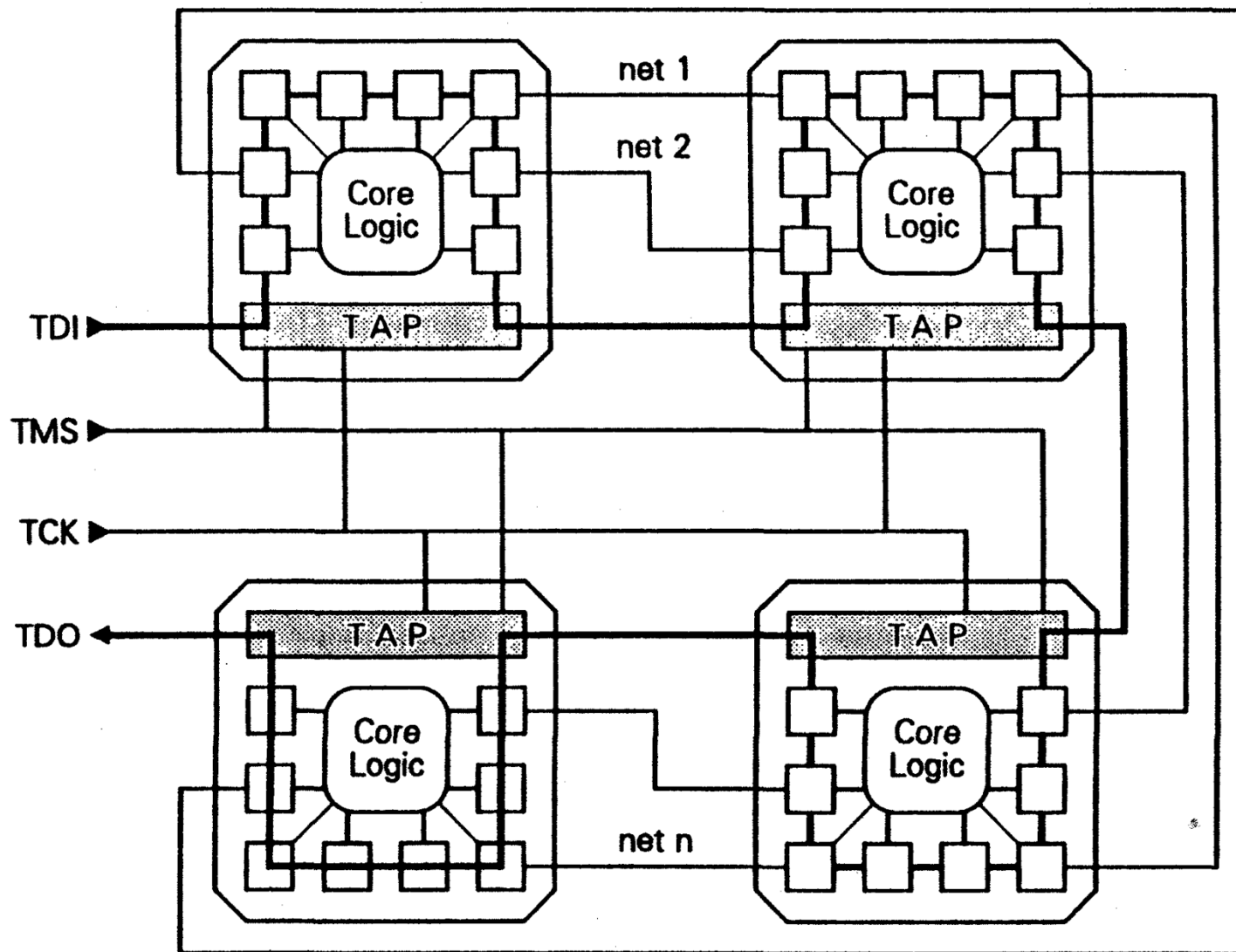
Инфраструктура BS-тестирования в ИМС



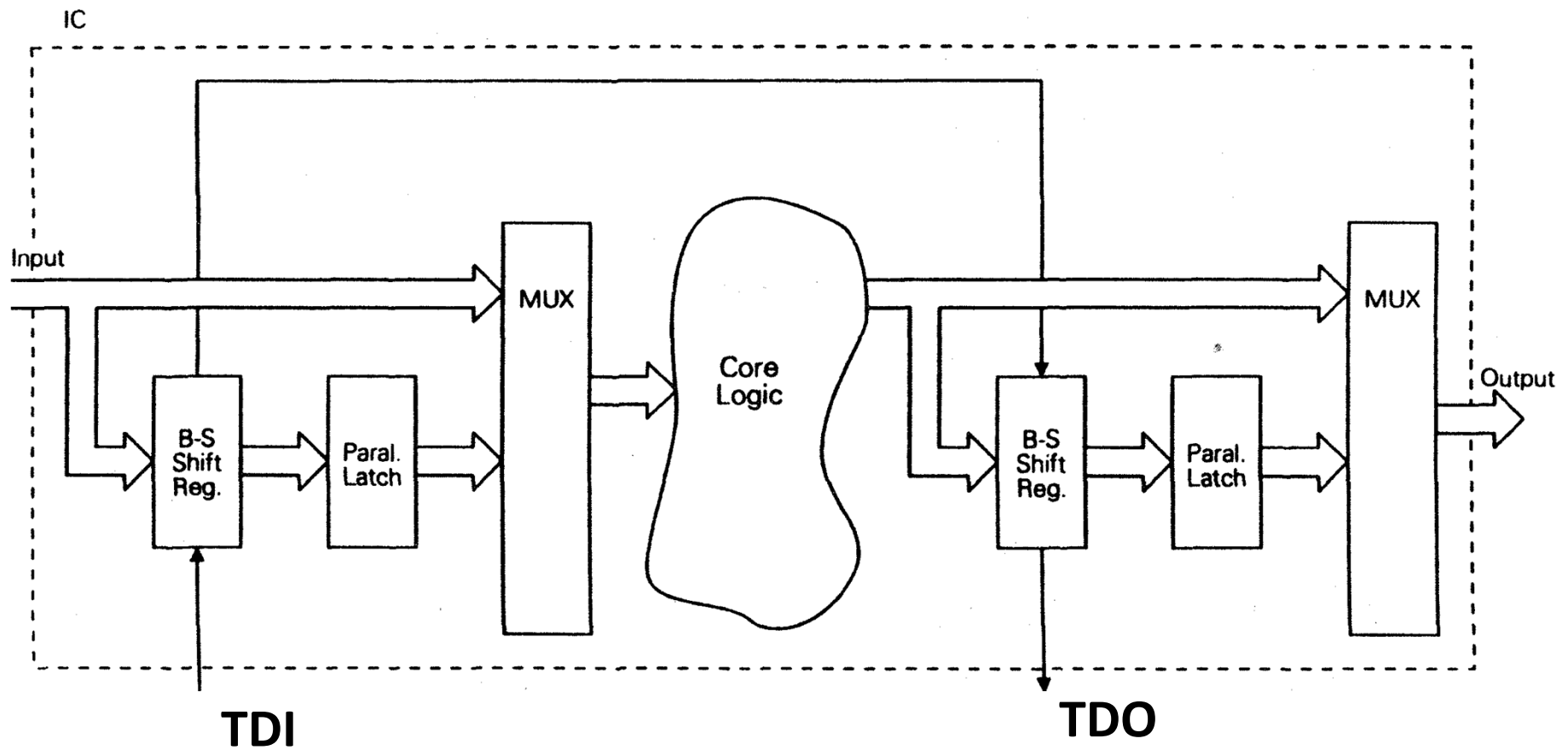
Инфраструктура BS-тестирования на плате



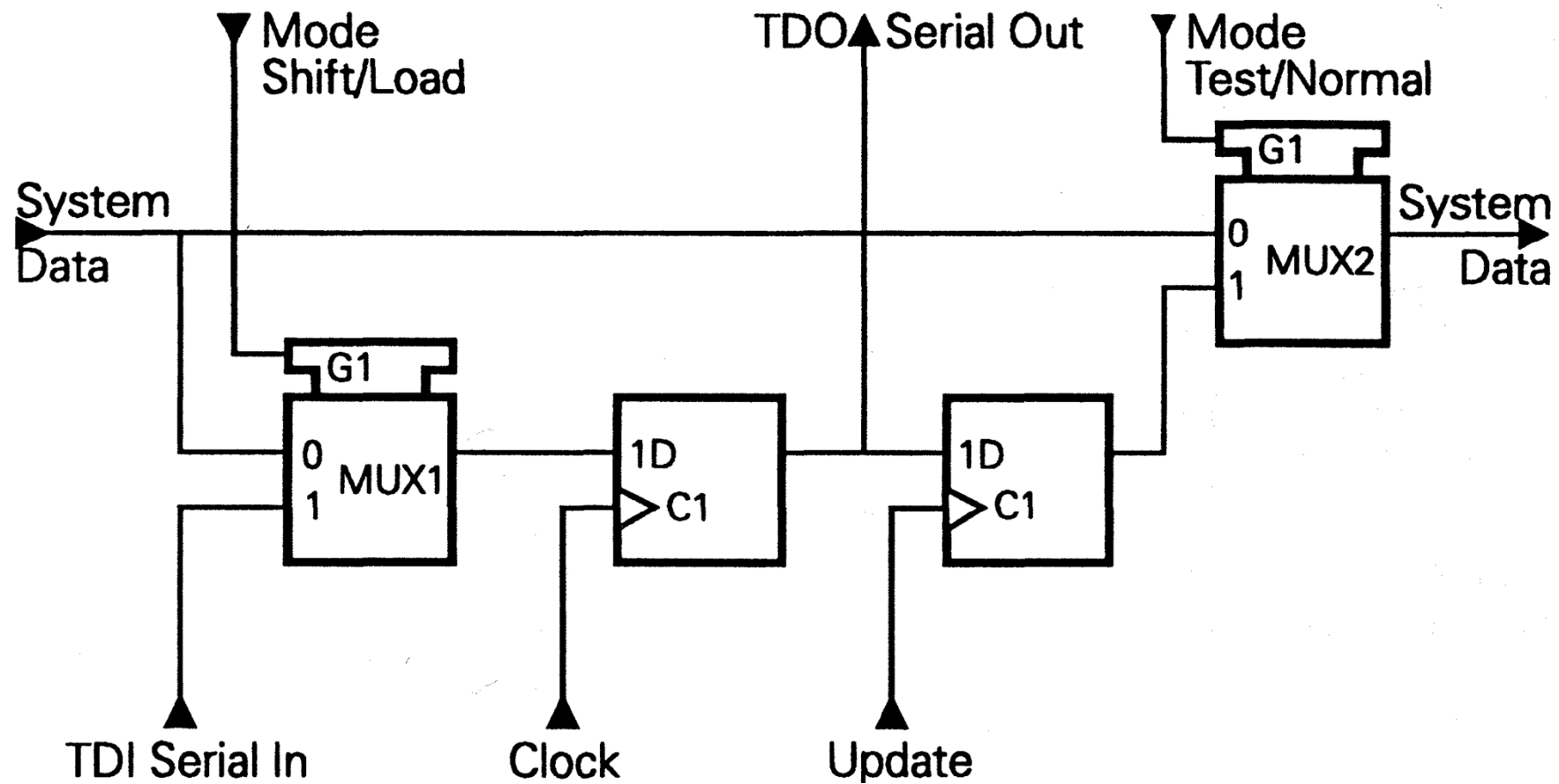
Инфраструктура ВS-тестирования на плате с несколькими микросхемами



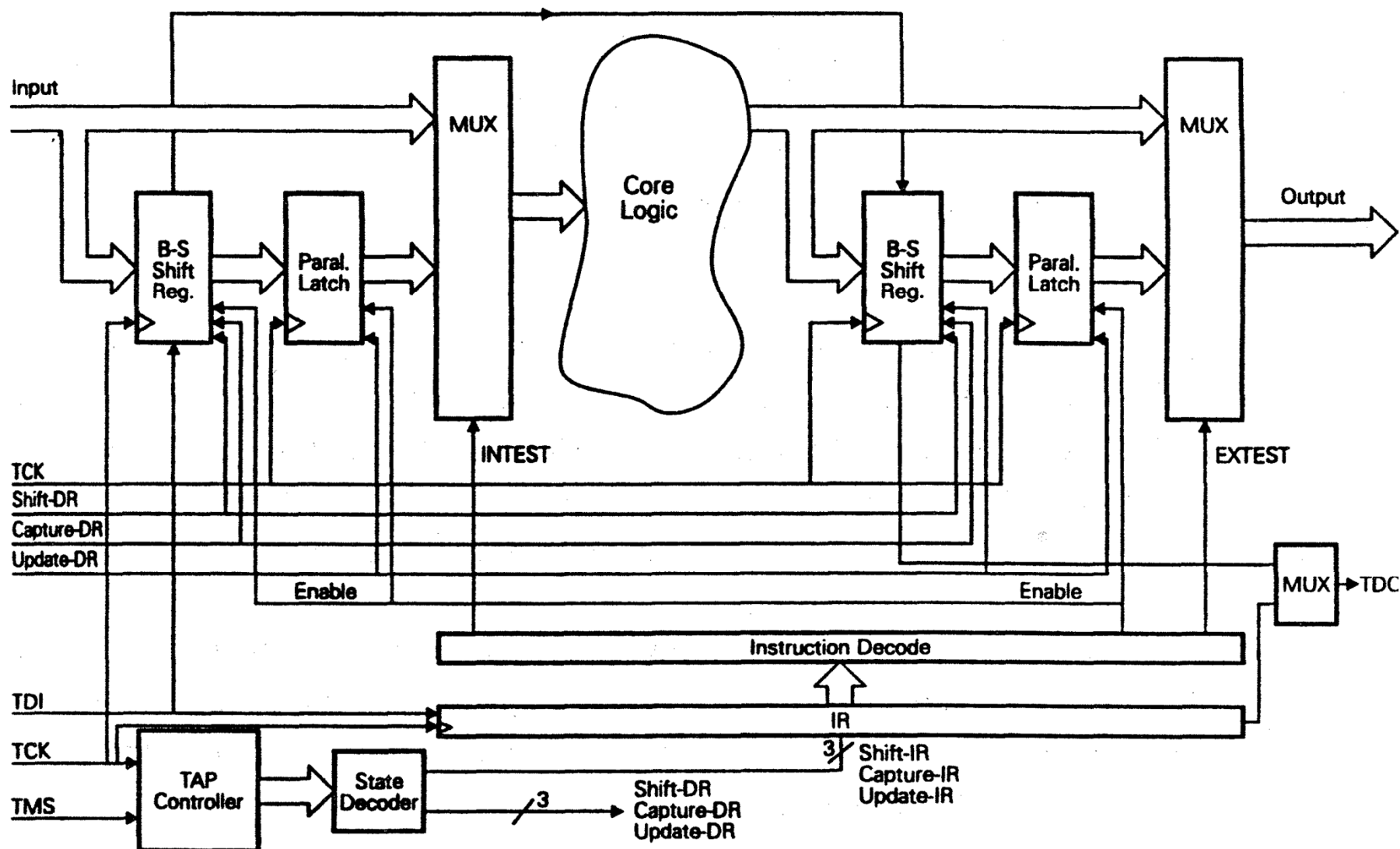
Функциональная схема ячейки граничного сканирования (Boundary-Scan Cell, BSC)



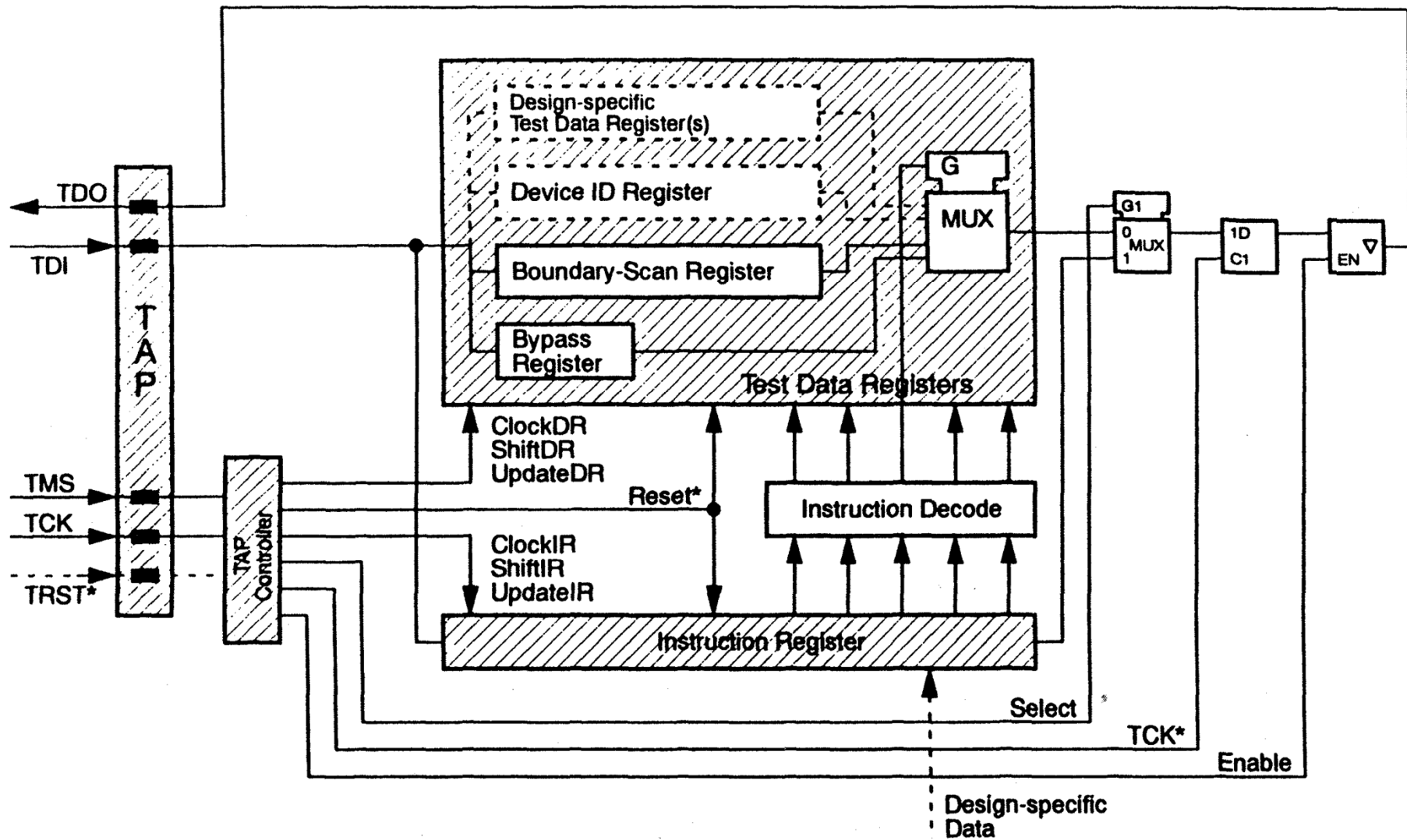
Принципиальная эл.схема ячейки граничного сканирования (Boundary-Scan Cell, BSC)



Управление BS-тестированием



Cxema TAP (TAP - Test Access Port)



Сигналы TAP

TDI - порт последовательного ввода данных и команд

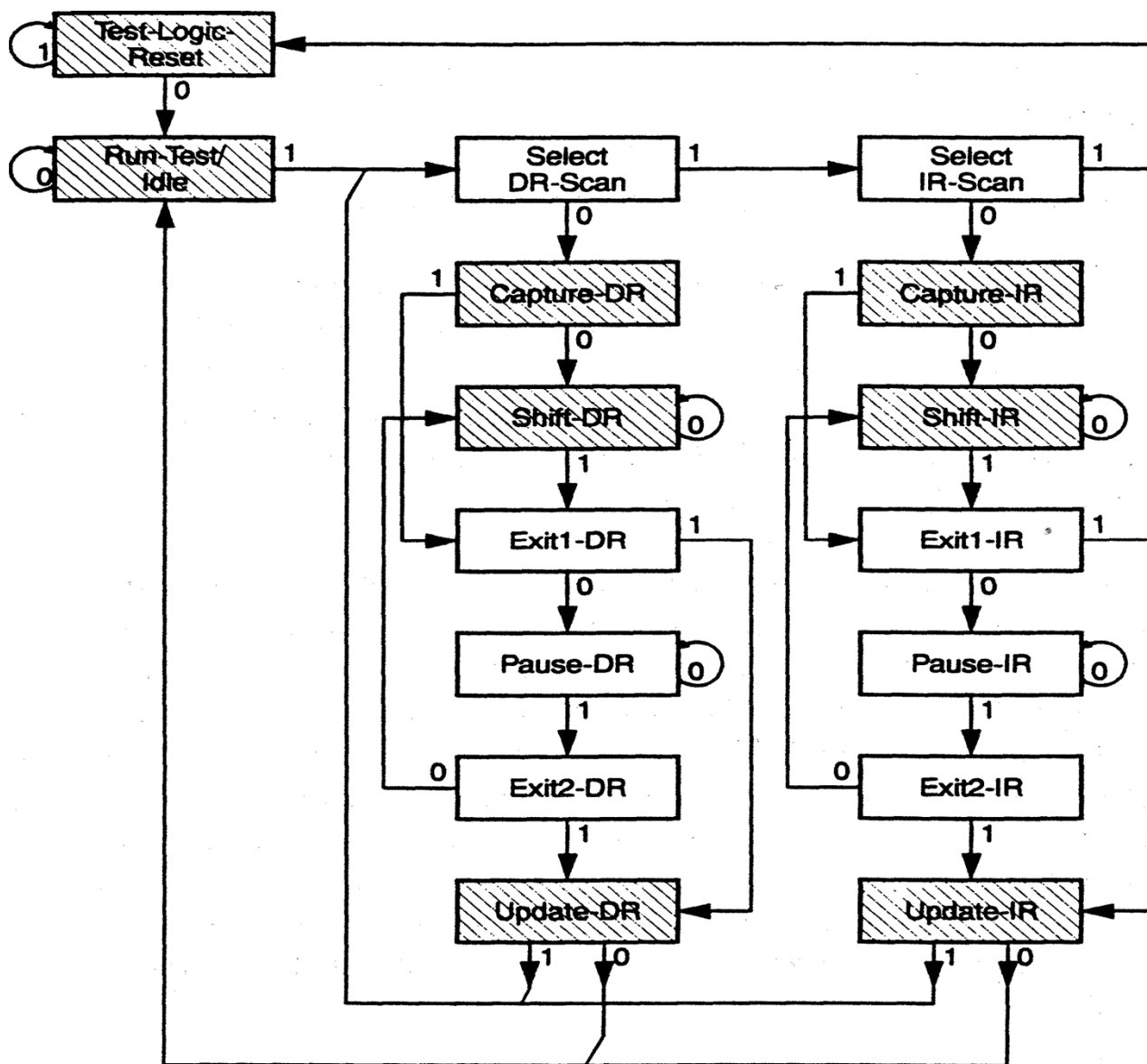
TDO - порт последовательного вывода и состояния

TMS - порт управления переключениями состояний TAP-контроллера

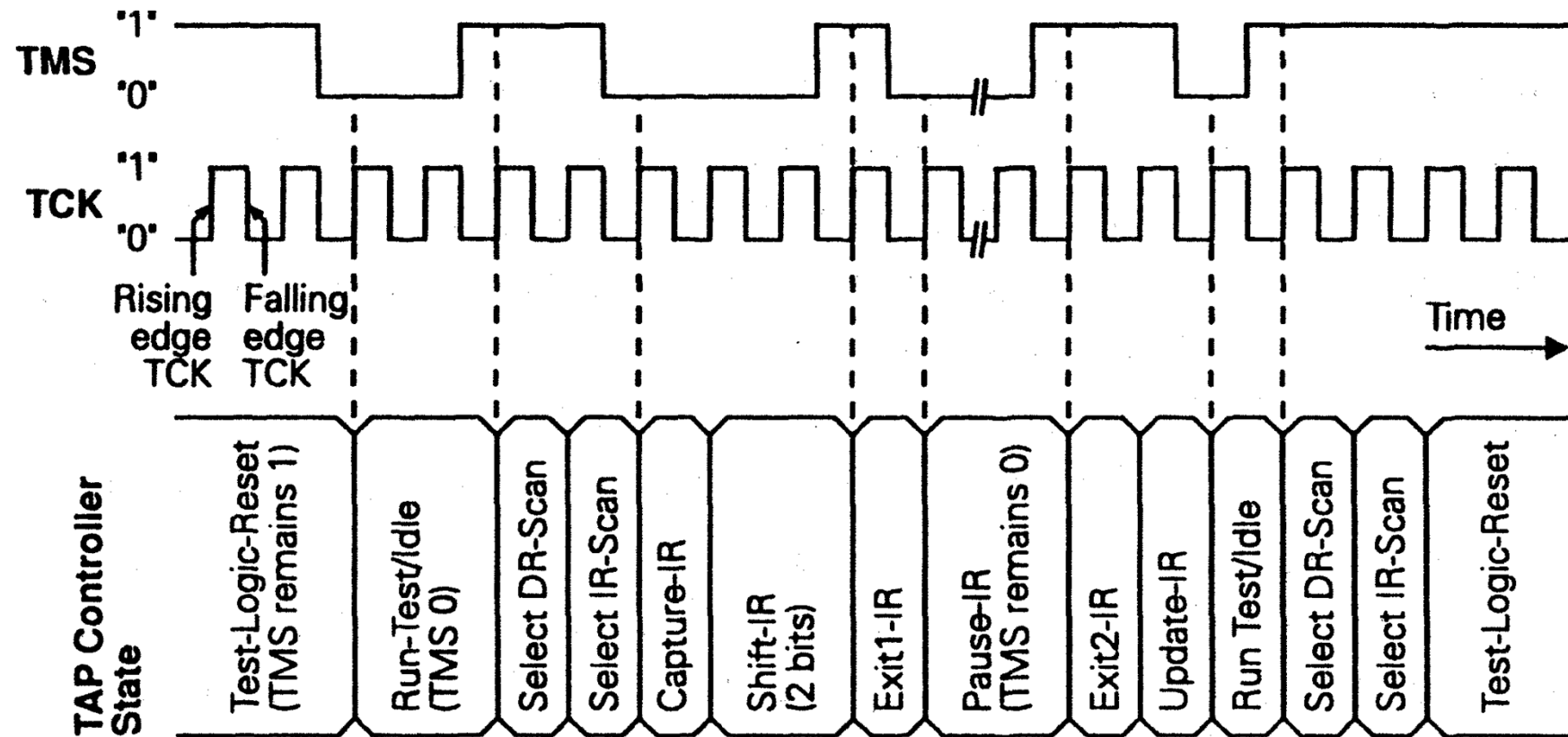
TCK - порт тактирования TDI, TDO, TMS

TRST - сброс состояния TAP-контроллера

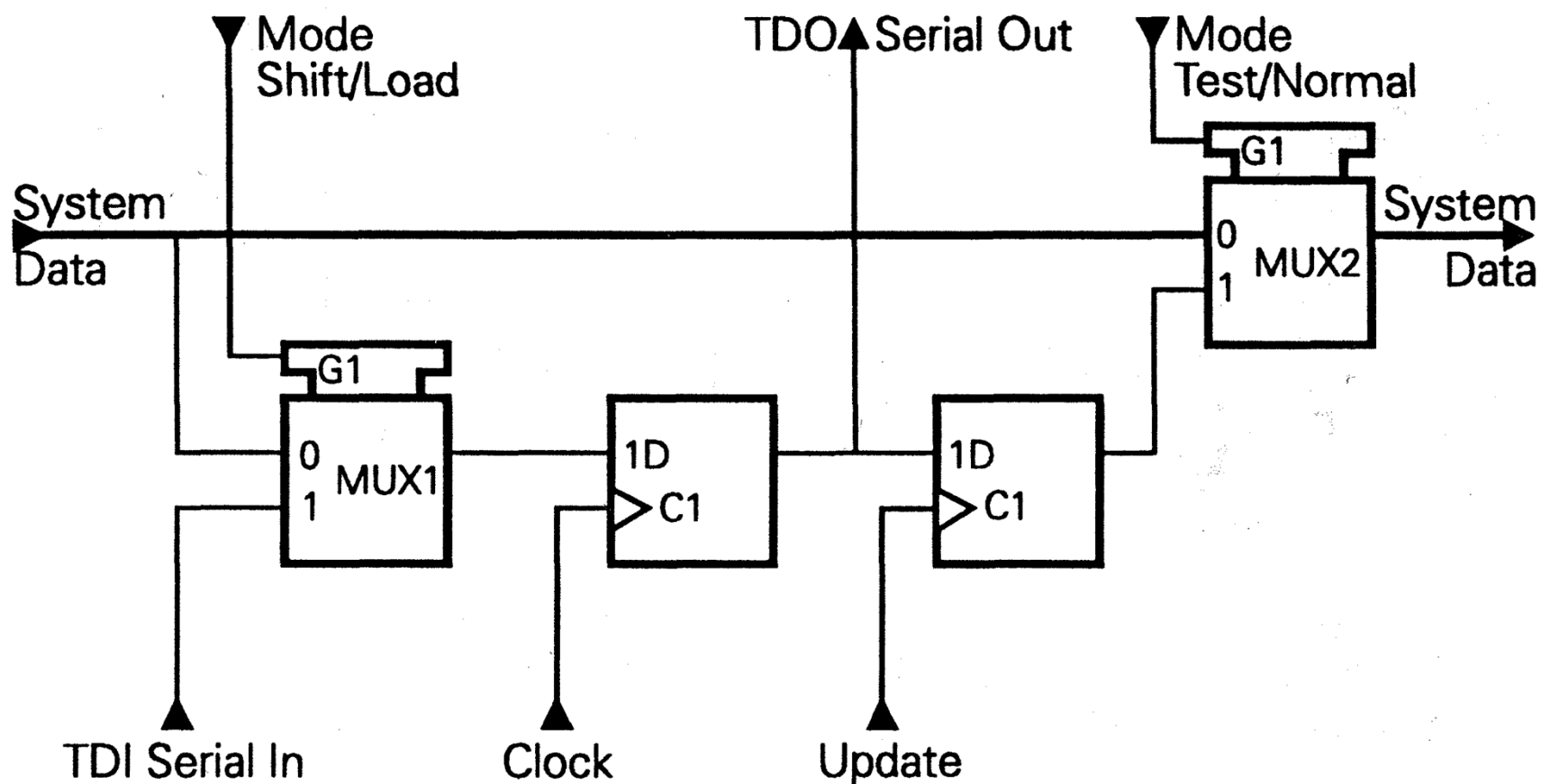
Диаграмма состояний ТАР-контроллера



Временная диаграмма TAP-контроллера

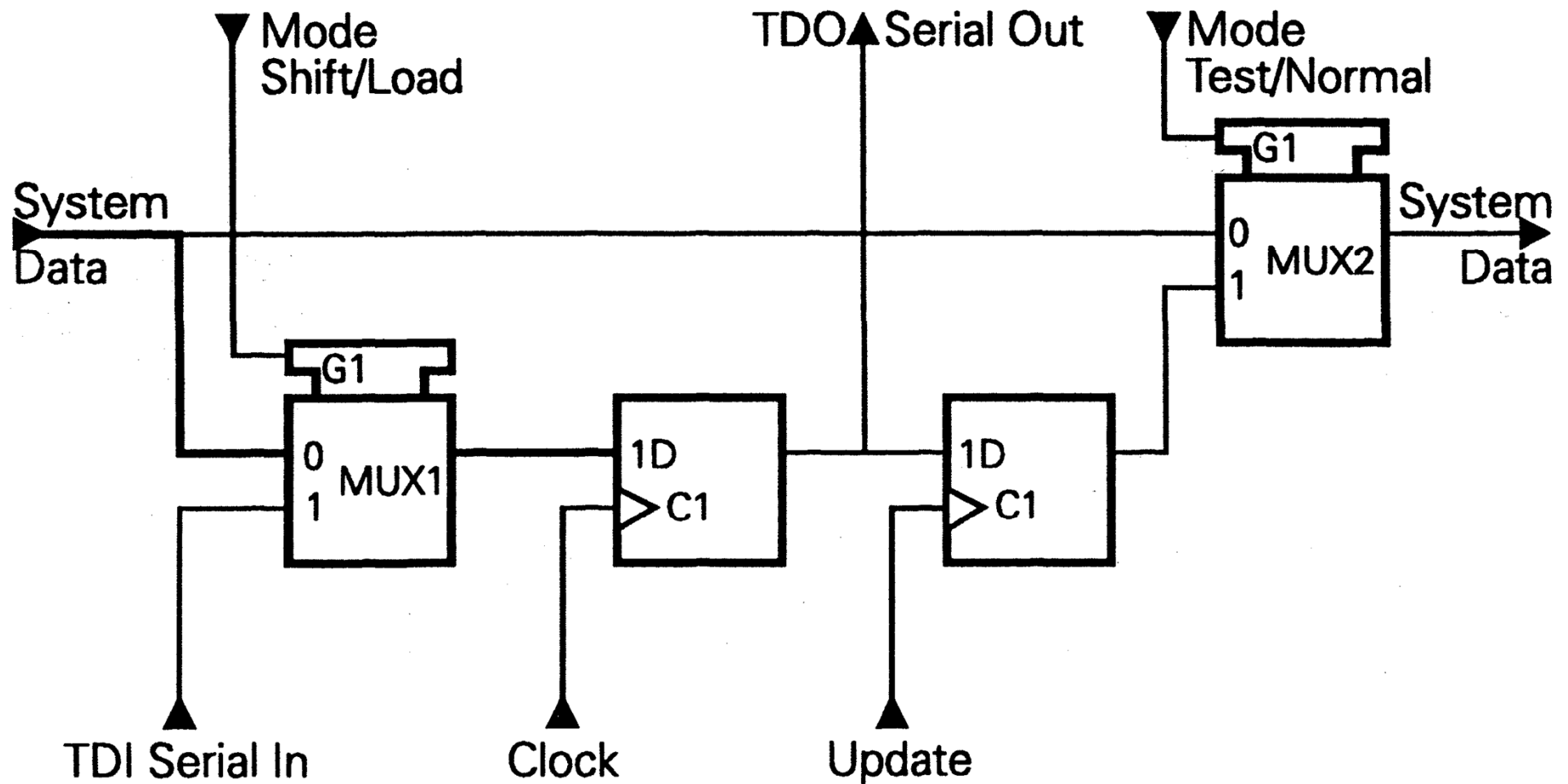


BS-ячейка в состоянии Test-Logic-Reset



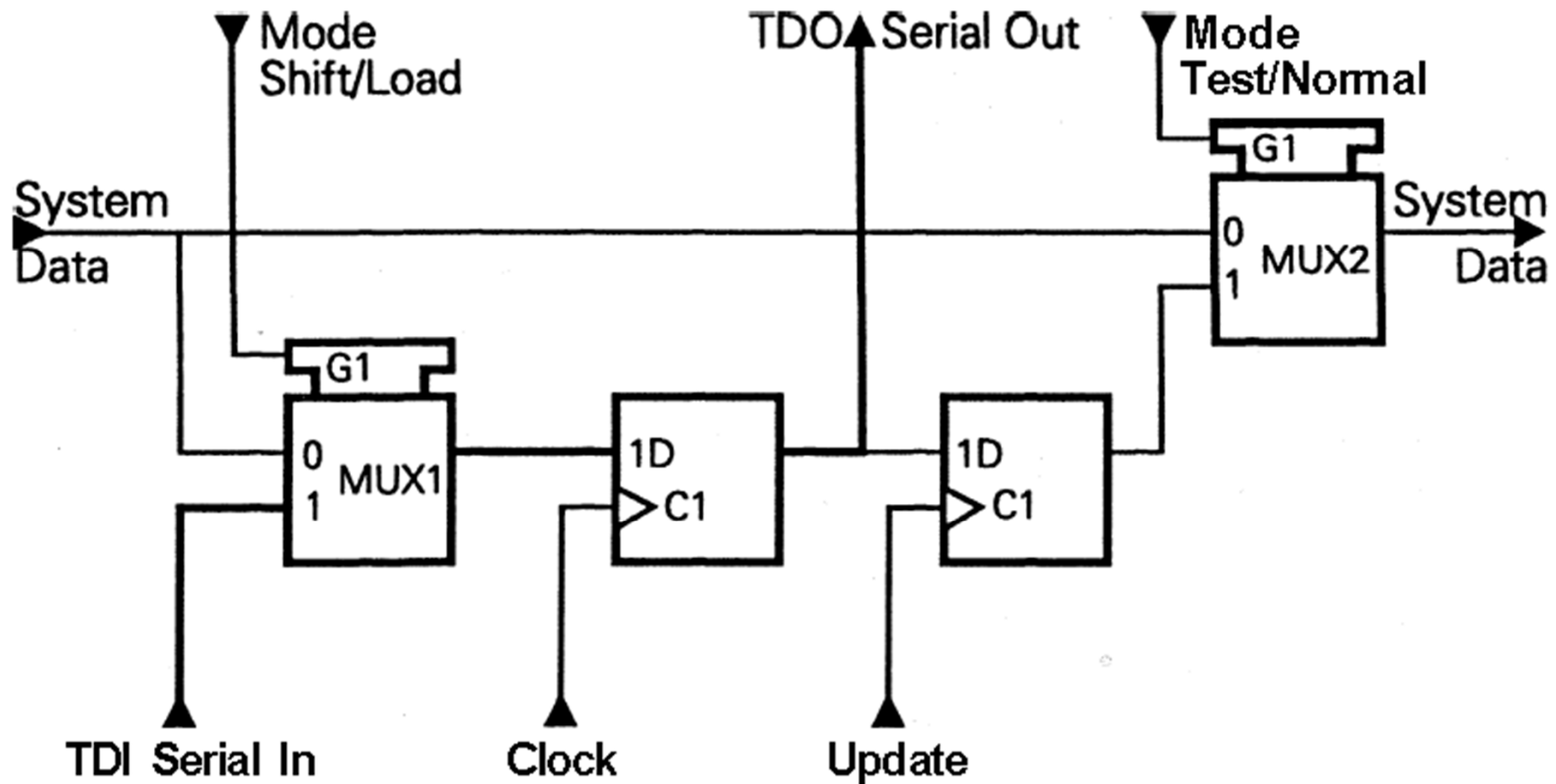
Прохождение сигналов выделено толстыми линиями

BS-ячейка в состоянии Capture-DR



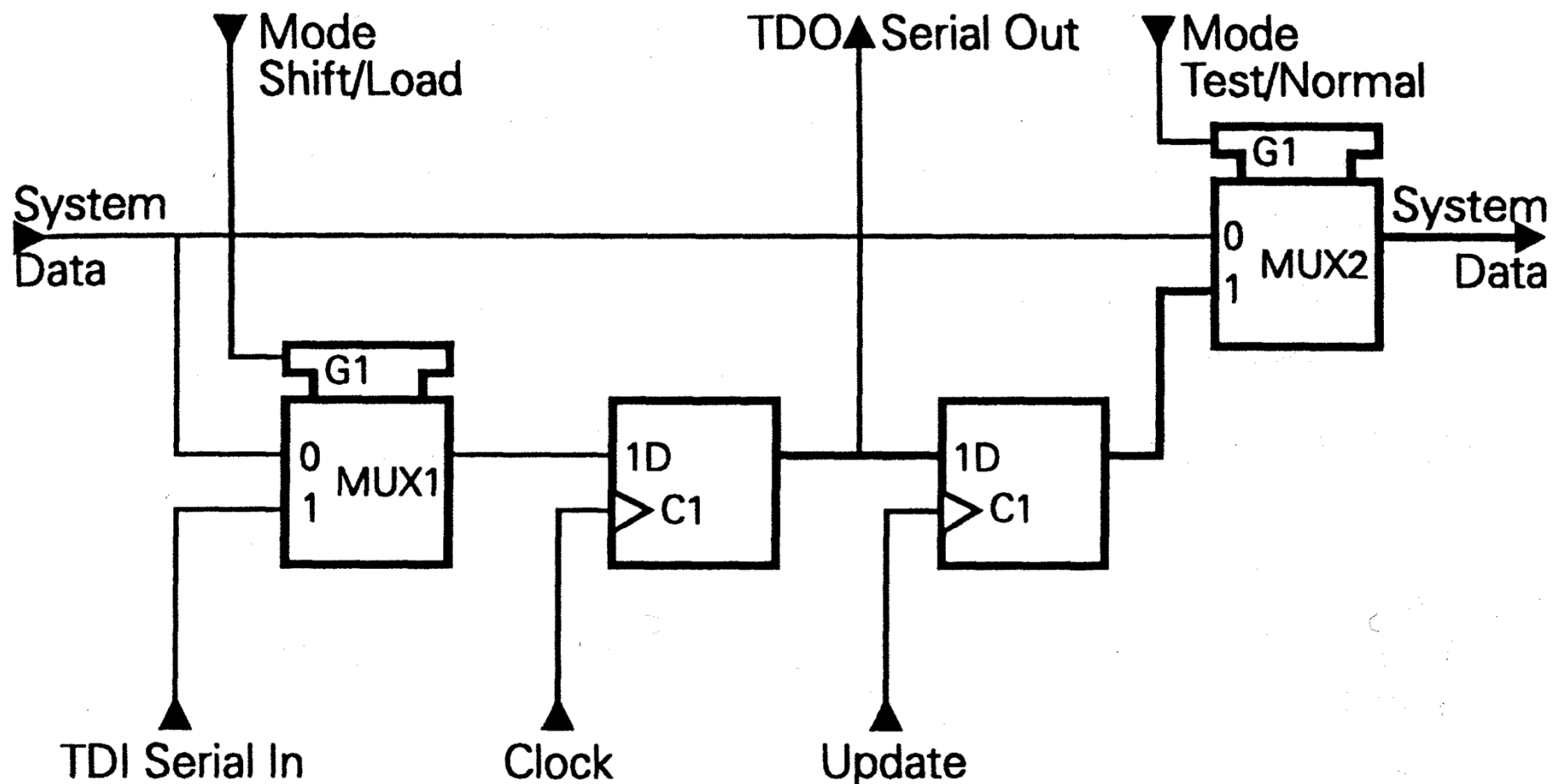
Прохождение сигналов выделено толстыми линиями

BS-ячейка в состоянии Shift-DR



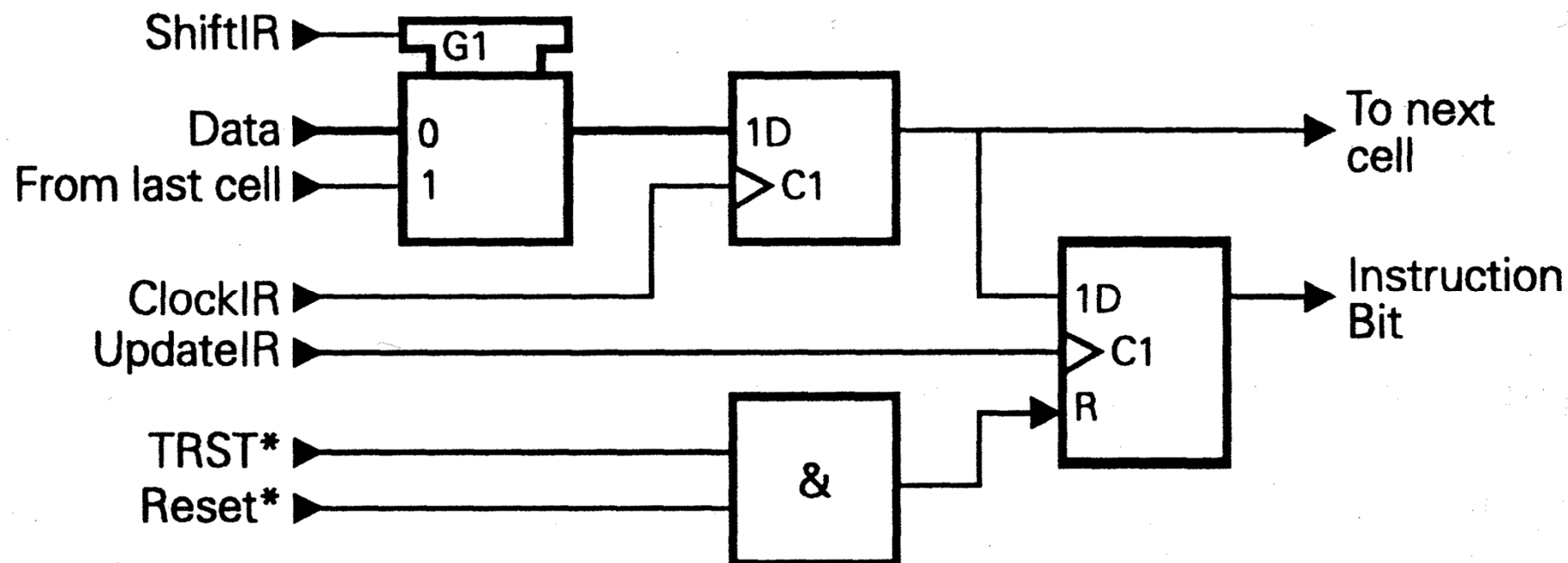
Прохождение сигналов выделено толстыми линиями

BS-ячейка в состоянии Update-DR



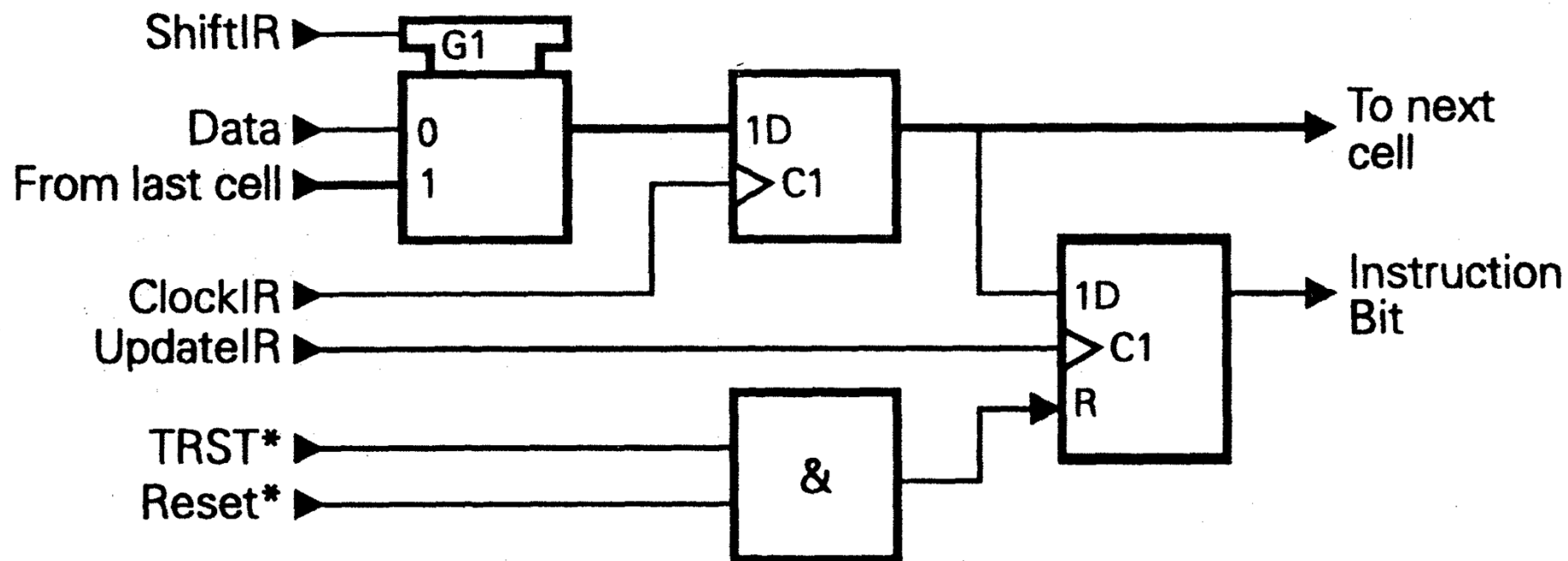
Прохождение сигналов выделено толстыми линиями

BS-ячейка в состоянии Capture-IR



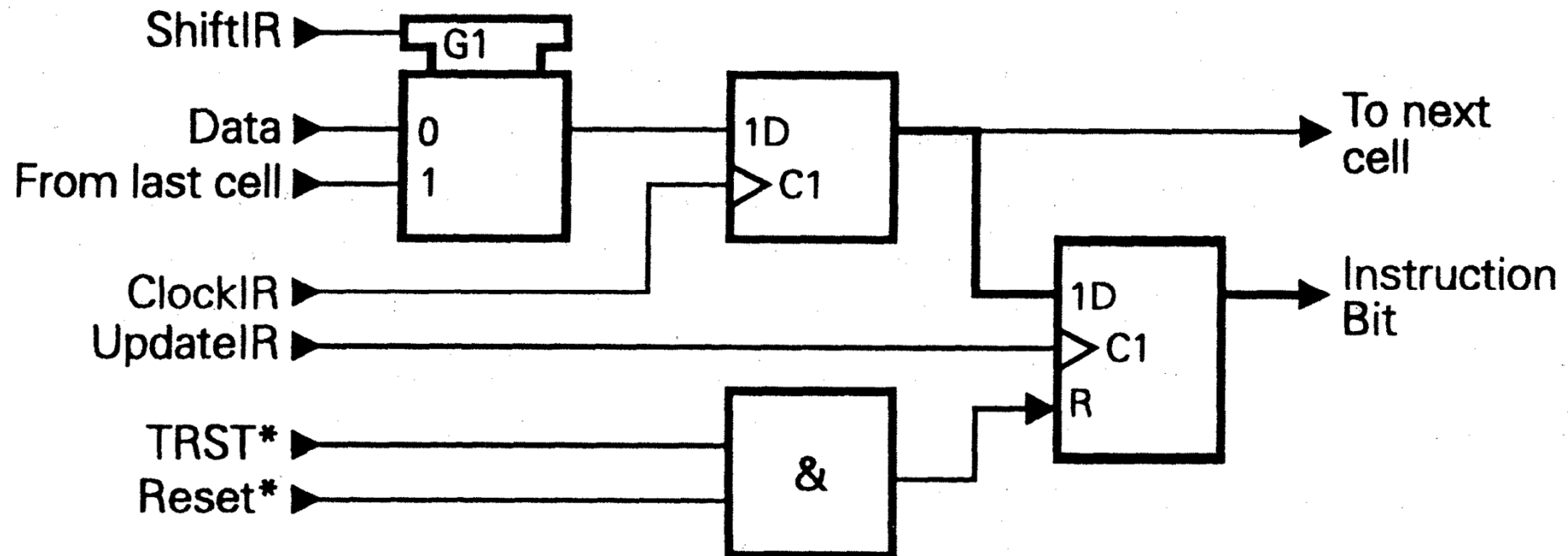
Прохождение сигналов выделено толстыми линиями

BS-ячейка в состоянии Shift-IR



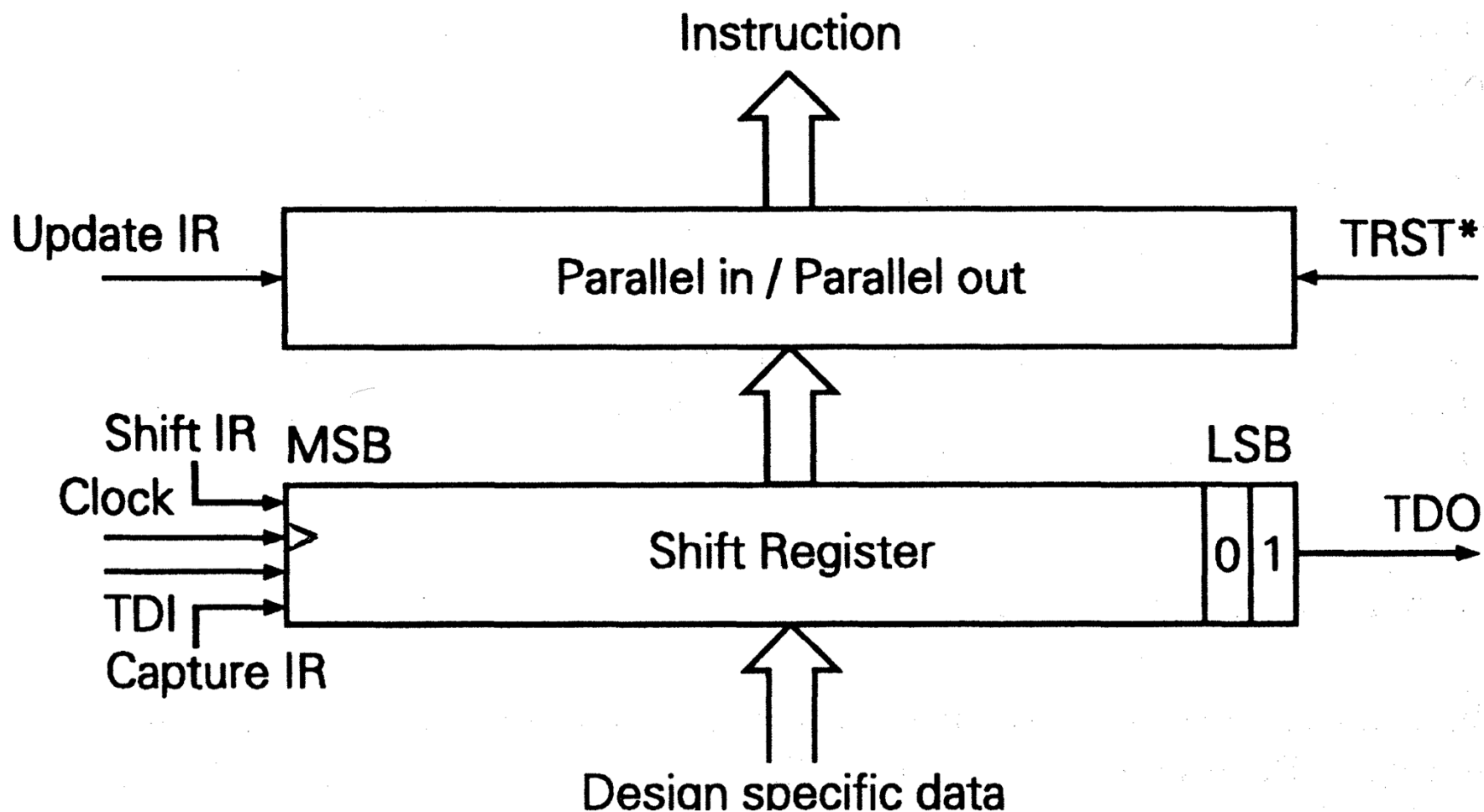
Прохождение сигналов выделено толстыми линиями

BS-ячейка в состоянии Update-IR



Прохождение сигналов выделено толстыми линиями

Запись регистра команд (Update-IR)



Прохождение сигналов выделено толстыми линиями

Порядок обновления регистра команд IR

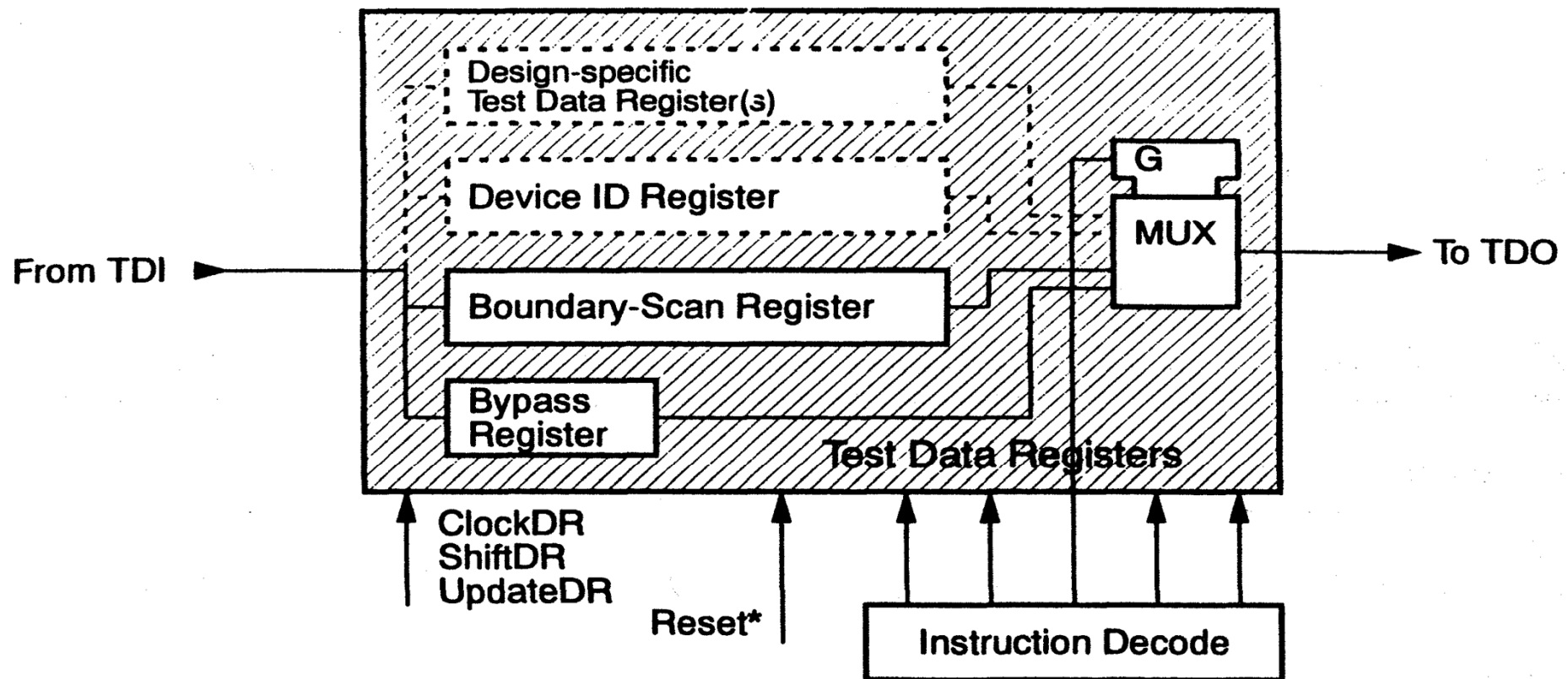
Состояние TAP-контроллера	Содержимое сдвигового регистра (Shift-Register)	Содержимое регистра команд (выход Instruction)
Test-Logic-Reset	Не определено	Выбор регистра IDCODE (или BYPASS)
Capture-IR	Загрузка 01 в младшие биты и иных битов состояния (специфика проекта)	Остается предыдущее состояние
Shift-IR	Вывод на выход TDO	Остается предыдущее состояние
Exit1-IR	Остается предыдущее состояние	Остается предыдущее состояние
Exit2-IR	Остается предыдущее состояние	Остается предыдущее состояние
Pause-IR	Остается предыдущее состояние	Остается предыдущее состояние
Update-IR	Остается предыдущее состояние	Загрузка команды из сдвигового регистра
Другие состояния	Не определено	Остается предыдущее состояние

Регистры (тестовых) данных (Test Data Registers, TDR)

Обязательные стандартные регистры: Boundary-Scan Register, Bypass Register

Оptionальные стандартные регистры: Device ID Register

Оptionальные специальные регистры: регистры, предусмотренные разработчиком ИМС



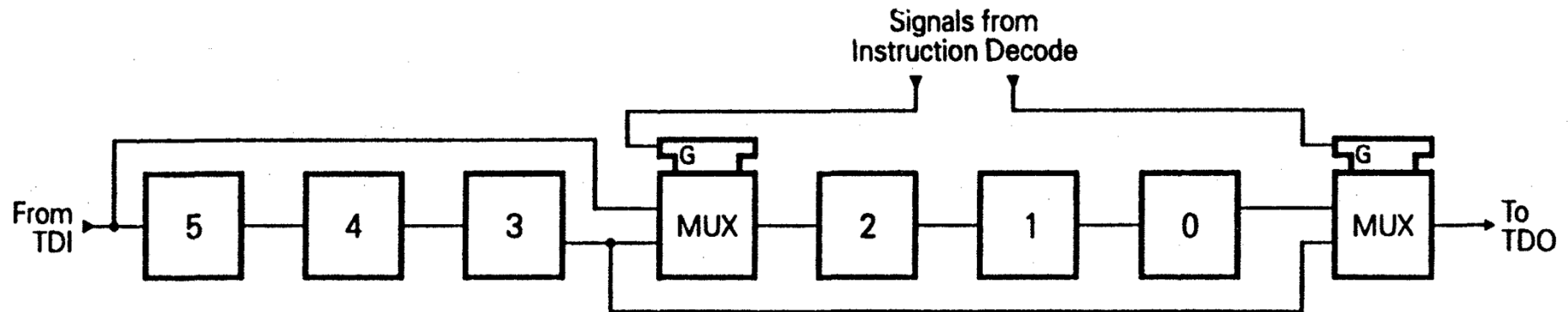
Выбор TDR осуществляется записью соответствующего кода операции в IR

Только для работы с BYPASS-регистром предусмотрен стандартный код: IR={111111...}

Коды IR для работы с другими регистрами выбираются разработчиком микросхем

Комбинированный регистр данных

Регистр данных может быть доступен по частям или целиком. Но при этом для каждого варианта доступа должна быть предусмотрена отдельная команда (записывается в IR)



Пример:

для 3-х различных значений IR (2 бита) могут быть выбраны 3-х битовые поля {5-4-3} ; {2-1-0} или 6-ти битовое поле {5-4-3-2-1-0},
которые логически доступны как 3 независимых регистра с различными кодами команд

Инструкции

- BYPASS – «обход» тестовой логики через однобитовый регистр
- SAMPLE/PRELOAD – тест без нарушения связей выводов с ядром ИМС.
- EXTEST – тест внешних (по отношению к ИМС) связей
- INTTEST (опц.) – тест ядра ИМС
- RUNBIST (опц.) – запуск внутреннего теста
- CLAMP (опц.) – BYPASS с предварительной установкой на выходах заданных уровней
- IDCODE/USERCODE (опц.) – считывание идентификаторов производителя и пользователя
- HIGHZ (опц.) – переводит все выводы в Z-состояние

IEEE Std 1500 - 2005 (IEC 62528-2007)

Standard Testability method for Embedded Core-based Integrated Circuits

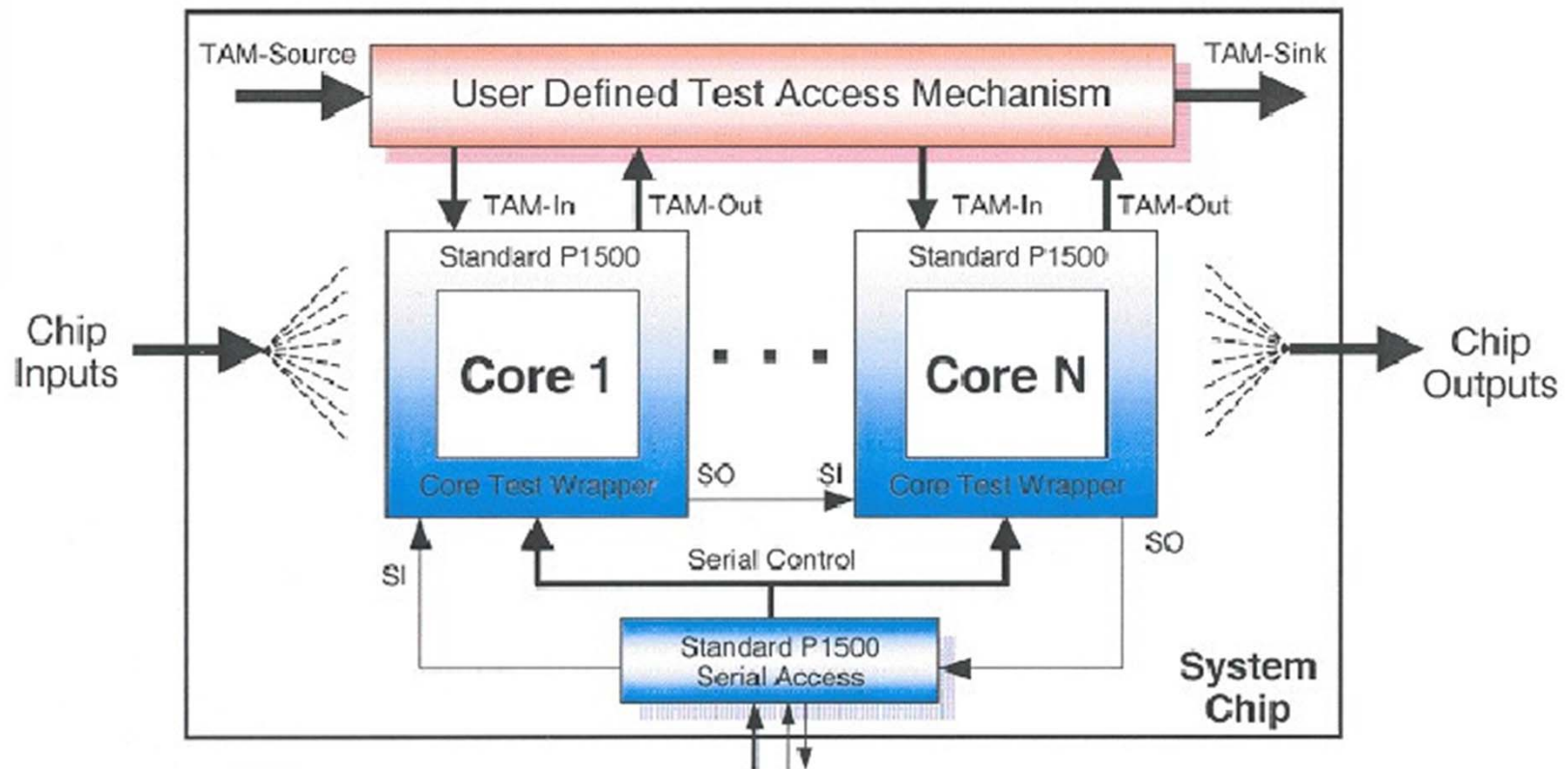
- Определяет метод тестирования СнК, построенных на основе IP-ядер (IP-Core)
- Определяет интерфейс тестирования между СнК (микросхемой) и IP-ядрами внутри этих СнК
- Обеспечивает повторное использование разработанных индивидуальных тестов IP-ядер уже в составе СнК за счет механизмов доступа и изоляции IP-ядер

Базируется на архитектуре IEEE 1149.1

Чем не подходит IEEE 1149.1 для тестирования IP-ядер?

- Требуются дополнительные порты (например, для поддержки Transfer Mode или AUXCK)
- TAP-контроллер должен быть модифицирован под каждое IP-ядро в СнК, даже для одного семейства СнК. Не допускается нескольких TAP-контроллеров.
- Невозможен одновременный (параллельный) доступ к портам СнК и портам IP-Core
- Фиксированные длины Bypass-, ID- и IR-регистров не всегда эффективны для IP-core.
- Тестирование ТОЛЬКО в «последовательном» режиме с ограниченной скоростью
- Слабо определены функции BIST
- Нет множественных CLOCK-сигналов
- Нет режима «внутреннего проигрывания» тестов (Transfer Mode)

Архитектура СНК по IEEE Std 1500



Режимы тестирования IEEE Std 1500

Normal Mode – нормальное функционирование СнК

Internal Core Test – внутренний тест IP-ядер

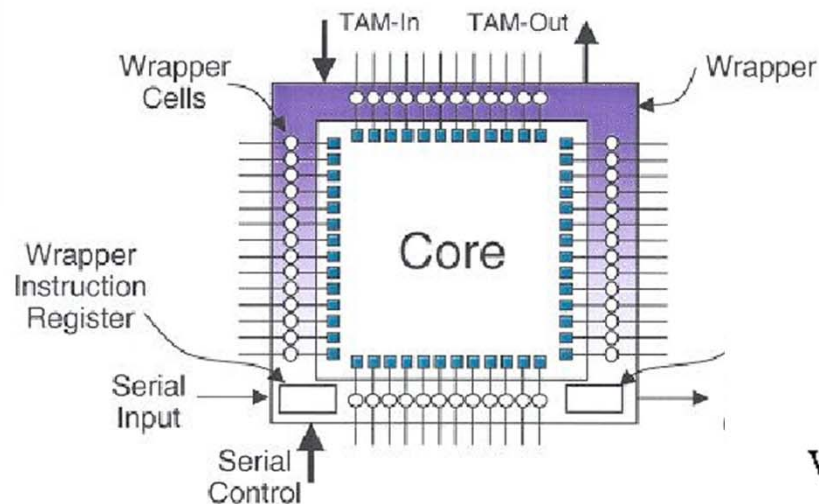
Core Interconnect Test – тест связей и логики между IP-ядрами в СнК

Test Isolation Mode – изоляция IP-ядер от СнК (отключение или установка «безопасных» уровней на выходах) для тестирования других IP-ядер или связывающей логики СнК

Функции тестирования, поддерживаемые IEEE Std 1500

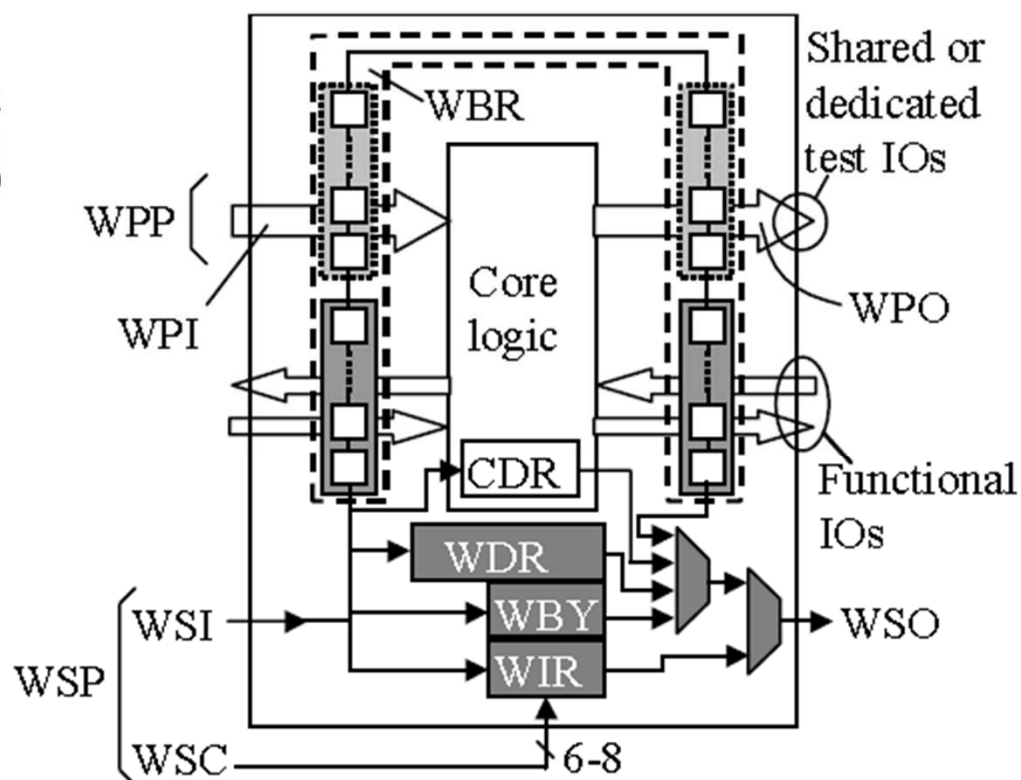
- Тестирование входов:
 - Наблюдение
 - Установка (тестовых значений «на вход»)
- Тестирование выходов:
 - Наблюдение
 - Установка (значений выходов)
 - Отключение

Структура тестируемого IP-ядра

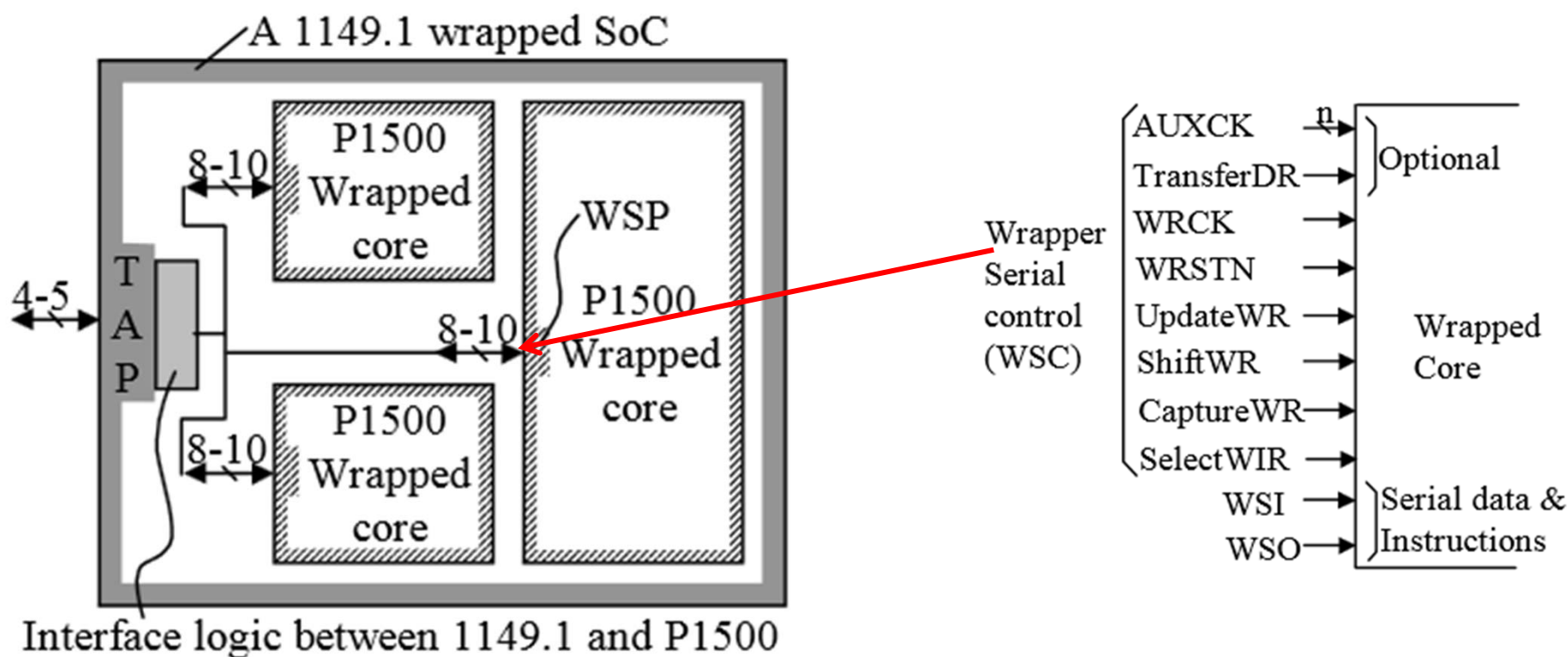


Обобщенная структура

Структурно-функциональная схема



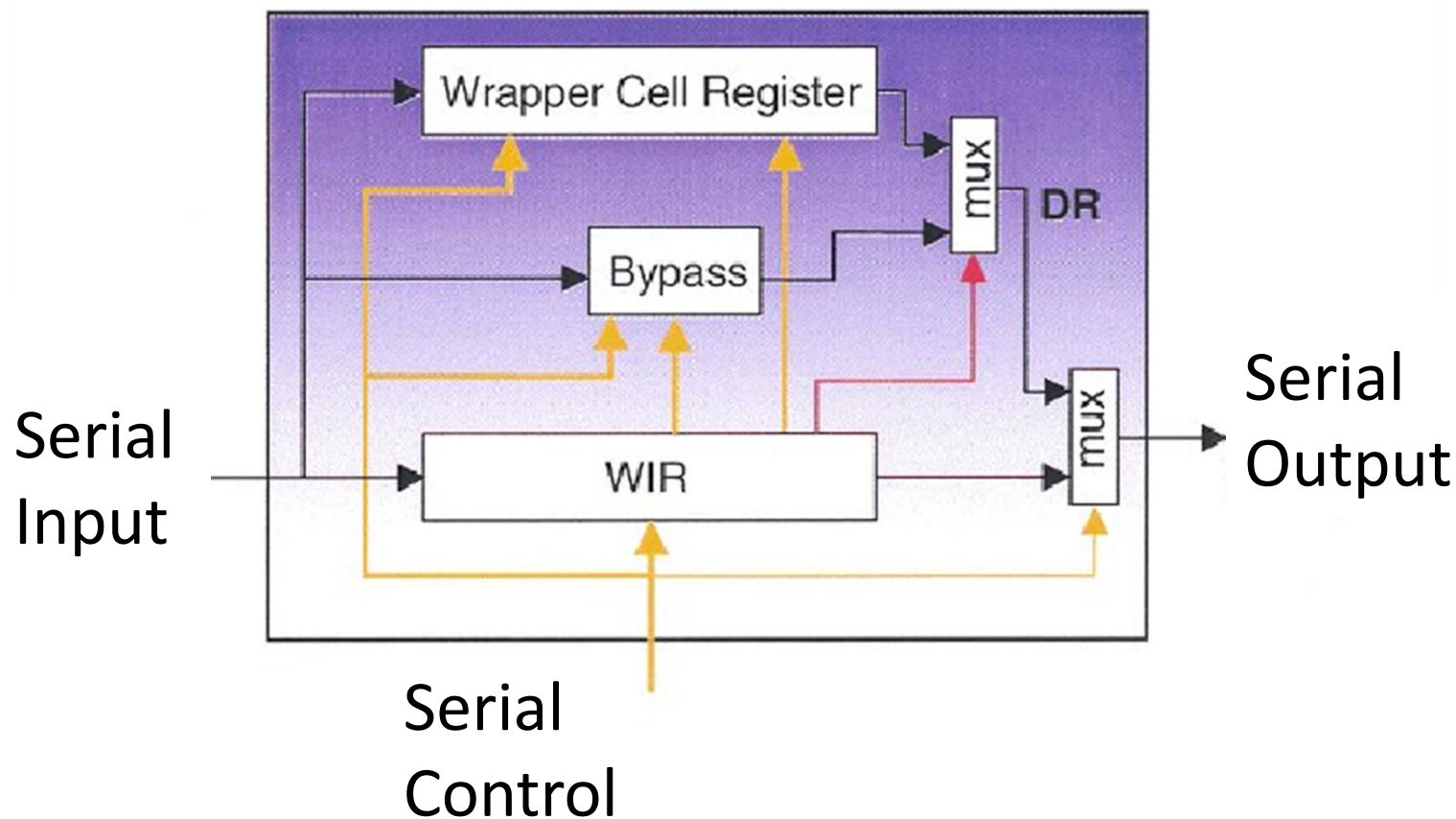
Интеграция и интерфейс IP-ядер



Стандарт IEEE 1500 не устанавливает однозначно способов интеграции с интерфейсом IEEE 1149.1

Сигналы WSC-интерфейса обеспечивают совместимость с интерфейсом IEEE 1149.1

Регистры «тестовой оболочки»



Wrapper Instruction Register – управляет режимами доступа/тестирования IP-ядра

Wrapper Cell Register – устанавливает/считывает состояния портов IP-ядра

Bypass – «обход» WCR

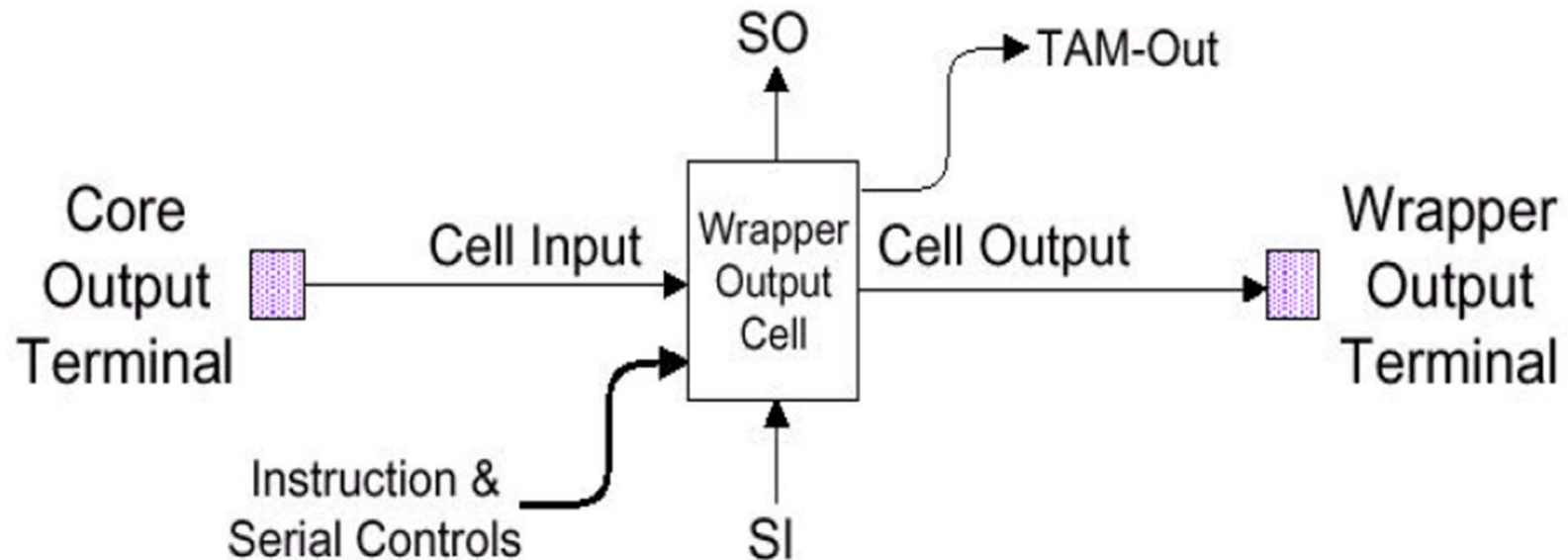
Обязательные команды (для WIR)

- WS_BYPASS
 - WS_EXTEST
 - Wx_INTEST (x = S(erial), P(arallel), H(ybride))
 - WS_PRELOAD
-

Режимы тестирования

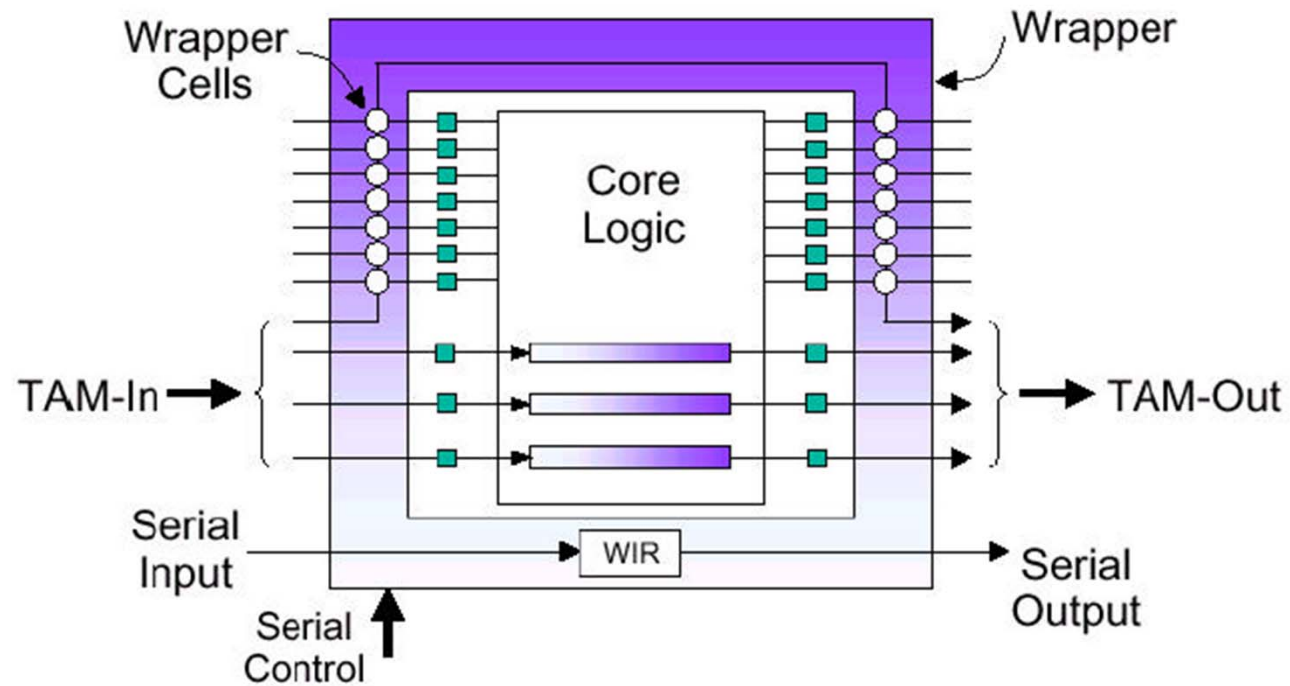
- Normal – наблюдение
- Core Test (1-N) – S,P,H-тестирование ядра
- Serial External Test – тестирование окружения IP-ядра
- Isolation – изоляция: Z-или фиксир. состояние

Режимы тестирования (2)



- Normal – CellOutput = CellInput
- Core Test (1-N) – S,P,H-тестирование ядра через TAM- или послед. порты
- Serial External Test – CellOutput устанавливается в стадии UPDATE
- Isolation – CellOutput = Z- или фиксир. состояние

Пример подключения TAM



Core Test Language (CTS)

- Эквивалент VSDL для ОЕФП
- Описывает тестирование в рамках IEEE Std.1500