

Моделирование асинхронной логики на переключательном уровне. II

Андрюхин А.И., Клименко И.В.
Кафедра ПМиИ, ДонНТУ
alexandruckin@rumbler.ru

Abstract

Andruckin A.I., Klimenko I.V. Simulation of Asynchronous Logic on switch level. Modeling of basic primitives of Asynchronous Design on switch level are considering. The results of simulations are presented.

Введение

Известны преимущества асинхронных схемы по сравнению с синхронными [1-4]. Однако сложность проектирования асинхронных схем является одной из основных причин доминирования синхронных схем.

Задача кодирования состояний является одной из основных задач канонического метода структурного синтеза автоматов. Переход автомата из одного состояния в другое осуществляется за счет изменения состояний элементов памяти. При функционировании автомата могут появиться так называемые состязания (гонки). Явление состязаний возникает вследствие того, что элементы памяти имеют различные, хотя и достаточно близкие, времена срабатывания. Кроме того, различны также задержки сигналов возбуждения, поступающих на входные каналы элементарных автоматов по логическим цепям неодинаковой длины

При кодировании состояний гонки должны быть устранены. Имеются 4 способа устранения гонок: двойная память; рациональный выбор длительности синхроимпульса; развязывание пар переходов; соседнее кодирование.

Как правило, чтобы гарантировать правильное выполнение операций, когда разработана асинхронная схема управления типа AFSM (Асинхронный Конечный Автомат - Asynchronous Finite State Machine), она должна быть протестирована на наличие паразитных импульсов и скачков переменной состояния (фазовой переменной). После выполнения теста, возможно, нужно добавить дополнительные схемы, для устранения комбинационных паразитных импульсов, должны быть идентифицированы критические пути, для устранения возможностей последовательных появлений паразитных импульсов, и новые фазовые переменные для устранения скачков. Кроме того, должно быть под контролем поведение оборудования (окружающей среды), т.е. его быстродействие и количество

одновременно изменяемых входных сигналов. Большинство асинхронных схем нуждаются в single-input-change и основном режиме функционирования. Первое подразумевает, что только одному входному сигналу позволено измениться перед изменением состояния схемы, тогда как под последним понимается, что быстродействие оборудования (окружения) должно быть меньше, чем быстродействие схемы.

Метод проектирования асинхронных схем

В [4] рассмотрен метод, использующий специальный тип переменной состояния (onehot)]. Каждая такая переменная и ее значение соответствуют каждому определенному состоянию конечного автомата. Этот метод устраняет скачки между переменными состояниями, т.к. он не кодирует состояния. Поскольку логическая схема, которая генерирует сигналы состояний, принимает регулярный вид, это упрощает ее создание.

Базируясь на one-hot методе кодирования, Холлар (Hollaar) предложил прямое создание one-hot асинхронных FSM, основанных на SR-триггерах [5]. Идея этой работы использовалась для успешного проектирования многочисленных схем управления при разработке асинхронного микропроцессора [6]. В [6] продемонстрирована технология прямой аппаратной реализации AFSM на КМОП ИС на основе комплекса динамических КМОП-примитивов, как показано на рис. 1. Каждая такая схема соответствует состоянию в AFSM и схемы состояния функционируют, как запоминающий элемент.

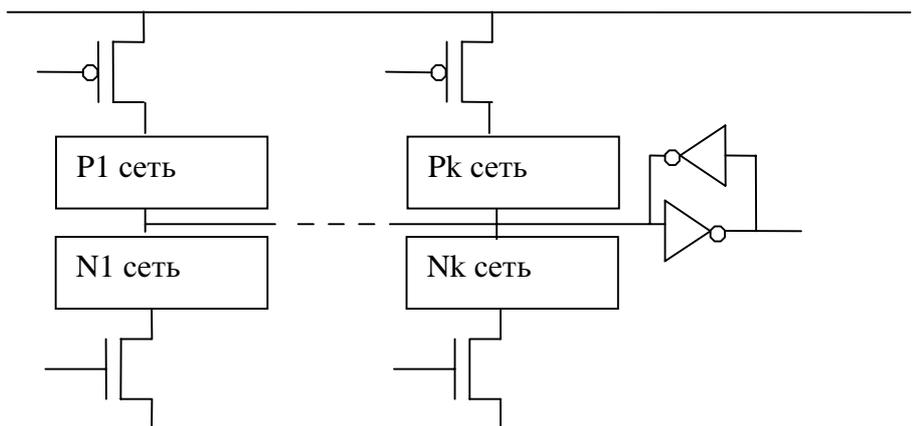


Рис.1. КМОП отображение состояний асинхронной схемы

Состояния FSM – групповые выходы таких схем. Когда выход схемы высокий или активный - тогда машина находится в соответствующем состоянии. Обычно, только одно состояние активно в любое время, хотя возможны параллельные FSM пути, где множество состояний активны одновременно.

В схеме состояний, $N_1,..N_k$ сети обнаруживают условия соответствующие входным состояниям, тогда как $P_1,..P_k$ сети обнаруживают условия выхода. $N_1,..N_k$ и $P_1,..P_k$ активизируются с помощью транзисторов. n -транзисторы, на которые подаются сигналы с предыдущих состояний, активизируют $N_1,..N_k$ сети, чтобы разгрузить схему состояний и таким образом запускают соответствующее состояние. p -транзисторы на которые подаются сигналы со следующих состояний, активируют $P_1,..P_k$ подсхемы, чтобы завершить состояние. Значение k зависит от количества состояний, от которых состояние может быть активировано. В самом простом случае, $N_1,..N_k$ могут соответствовать единственному n -транзистору, определяющему событие, что триггеры изменили состояние. Следовательно, два n -транзистора являются минимумом. Ясно, что вид $N_1,..N_k$ подсхем может быть более сложен в зависимости от условий для перехода в состояние. Соответственно, для p -транзисторов в самом простом случае схемой является коротким замыканием, т.е. просто подключаем инвертор к p -транзистору.

Когда, одно или более состояний предшествовало одному состоянию – то это состояния следования, т.е. есть один или более двойных циклов в диаграмме состояний. Для того чтобы гарантировать, что n -тип и p -тип части схемы никогда не **ВКЛЮЧЕНЫ** одновременно, необходимо ввести N_1, N_k -схемы,. Структура P_1, P_k -схем, в этом случае, должна быть такой же, как $N_1,.. N_k$ -схем следующего состояния, только с инвертированными вводами.

Для устранения one-hot критических скачков должно быть обеспечено, чтобы текущие и предыдущие изменения пар состояний находились в порядке $10 \rightarrow 11 \rightarrow 01$, т.е. следующее состояние должно быть введено прежде, чем текущее завершится. В этом методе, следующее состояние определяются его n -транзисторами, тогда как текущее состояние завершается его p -транзисторами. Для правильной работы схемы, разрешающий n -транзистор следующего состояния должен быть включен достаточно долго, до момента ввода следующего состояния. Этот разрешающий n -транзистор будет выключен p -транзисторами предыдущего состояния, как только сигнал переменной следующего состояния начинает расти. Для следующего состояния, которое будет введено, он должен оставаться включенным до момента переключения схемы состояний. Из этого следует, что верхние p -транзисторы на рис.1 должно медленнее переключаться, чем нижние n -транзисторы. Это может быть достигнуто установкой соответствующих размеров W и L (ширина и

длина затвора) транзисторов. На практике, обычно р-транзисторы существенно медленнее переключаются, нежели чем п-транзисторы.

Постановка задачи

Примитивы предложенного метода прямого отображения AFSM используют комплекс динамических КМОП-схем. Поэтому естественен переход на переключательный уровень для моделирования и тестирования устройств, использующих эти примитивы. Для этих целей используем описанный в [7-9] подход и модифицированную систему моделирования [10]. Результаты применения являются основной целью и содержанием данной работы.

Отображение графа состояний КМОП-примитивами

В результате прямого кодирования (представление каждого состояния конечного автомата с помощью одного своего триггера) в каждый конкретный момент времени, активным (hot) может быть только один триггер состояния. Прямое отображение AFSM является модульным, то есть может быть создано, как связанное множество AFSM сегментов. На рис.2 показано, как различные части AFSM могут быть реализованы соответствующими КМОП-примитивами.

На рис.2(а) показано, как может быть изображена часть графа состояний. Необходимо покинуть первое состояние, когда перешли в следующее введено. На рис.2(а), когда перешли в состояние $s3$, уже состояние $s2$ является покинутым. Это достигнуто с помощью р-транзистора для схемы $s2$.

Таб.1.Результаты моделирования схемы 2(а)

Номер набора	Число итераций	Значения сигналов в узлах схемы									
		3	4	5	6	7	8	9	10	11	12
1	6	D1	D1	D1	D1	D1	D0	D0	D0	D1	D0
2	5	D1	D0	D0	D1	D1	D0	D0	D0	D1	D0
3	5	D1	D0	D0	D1	D1	D0	D0	D0	D1	D0
4	4	D1	D1	D1	D1	D1	D0	D0	D0	D1	D0
5	8	D1	D0	D0	D0	D0	D1	D0	D1	D0	C0

На рис.2(б) показано, как может быть изображен цикл в графе состояний. Ввод z переводит машину в состояние $s1$. Если все три входа, которые образуют цикл, одновременно активны, то машина колеблется между тремя состояниями. Поскольку этот режим обычно неприемлем для FSM, должно быть гарантировано, что не менее чем два входа являются взаимоисключающими.

На рис.2(с) показана реализация варианта, в котором состояние имеет множество возможных приемников. Состояния $s2$ и $s3$, в этом примере, могут быть введены от состояния $s1$, вывод от состояния $s1$ подан к обоим из этих состояний. В случае параллельных путей в графе состояний входы, которые ведут FSM в ответвленные пути, должны быть взаимоисключающими, или что их значение должно являться исчерпывающим перед достижением ответвления состояний, это должно обеспечиваться окружающей средой. Последнее уместно в том случае, когда один вход - инверсия другого, для того, чтобы избежать последовательных интенсивных паразитных импульсов.

На рис.2(d) показано, как два параллельных пути в графе состояний объединяются в один. Поскольку состояние $s3$ – это состояние, где эти два пути объединяются и, следовательно, $s1$, и $s2$ - оба состояния-предшественники $s3$. это требует двух N1, N2 подсхем. По той же самой причине, на р-транзисторы для подсхем P1 и P2 состояний $s1$ и $s2$ будут поданы инверсии сигнала состояния $s3$.

Таб.2.Значения сигналов для схемы 2(d).

Номер Набора/ Число итераций	Значения сигналов в узлах схемы												
	3	4	5	6	7	8	9	10	11	12	13	14	15
	x	y				s3				s1	S2		
1/5	D0	D0	D0	D0	D0	D0	D1	CX	CX	D0	D0	D1	D1
2/7	D1	D1	D1	D1	D1	D1	D0	D0	D0	D1	D1	D0	D0
3/4	D1	D1	D1	D0	D0	D1	D0	D0	D0	D1	D1	D0	D0

Как было упомянуто, для правильного осуществления двойного цикла, P1,P2..схемы являются более сложными и пример представлен на рис.3.

Таб.3.Значения узлов для схемы на рис.3.

Номер набора/число итераций	Значения узлов схемы							
		w	x	y	z	S1	S2	S3
1/6	D1	D1	D0	D0	D0	D1	D0	D0
2/6	D0	D0	D1	D0	D0	C1	D1	D0

Интересно, что переход из состояния $s1$ в $s2$ при $x=1$ отражен в во 2 строке значением C1 для узла $s1$ (это емкостное значение слабее сигнала D0).

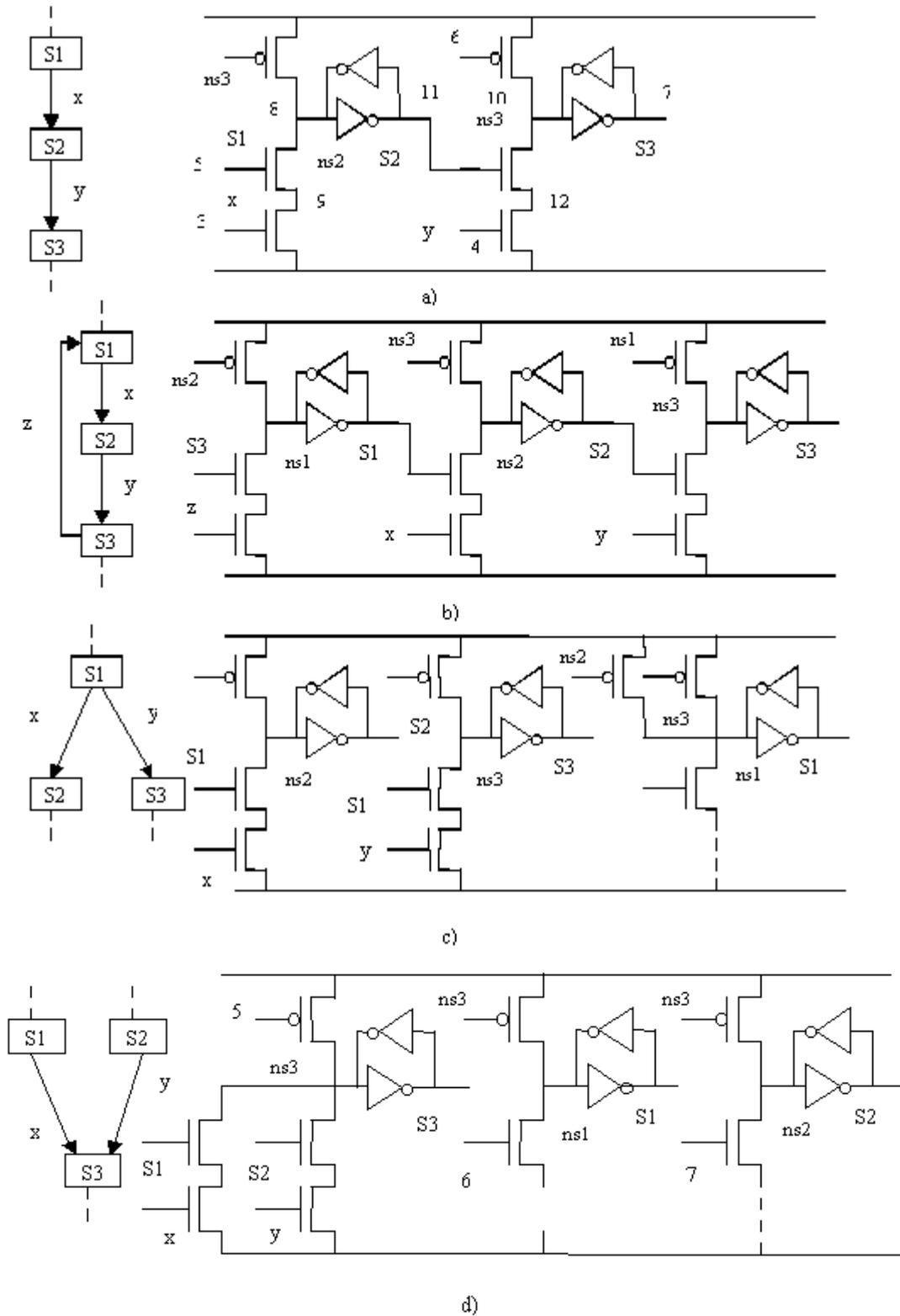


Рис.2. Отображение частей FSM графа состояний ИС на КМОП-примитивы: а) линейная последовательность; б) цикл; в) разветвление; г) слияние.

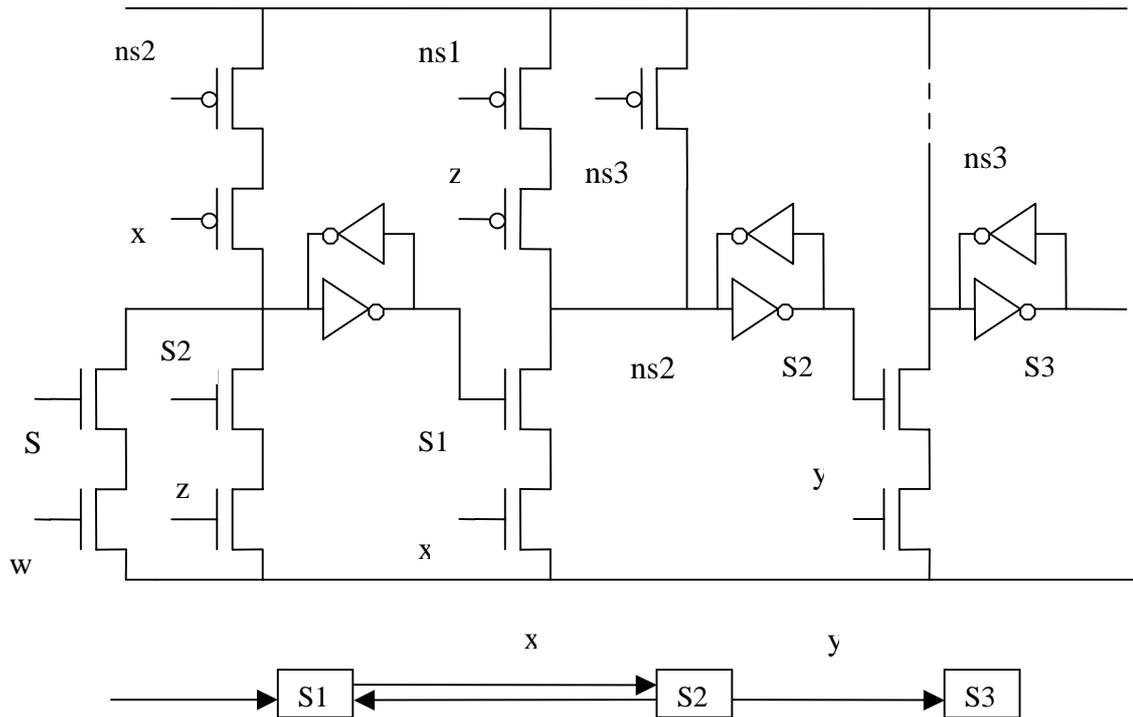


Рис.3.Двойная петля(цикл).

Заключение и перспективы дальнейших исследований

Реализуемые в настоящее время устройства с асинхронной логикой используют КМОП-примитивы, представленные на транзисторном уровне. Обоснованным является применение переключательного уровня при их моделировании. Моделирование примитивов асинхронной логики является первым шагом для их тестирования и тестирования устройств, в состав которых они входят. В настоящее время уже реализована возможность моделировать и строить проверяющие тесты для тех неисправностей, для которых это выполнять на вентиляльном уровне весьма затруднительно [11].

Литература

1. J.A.Brzozowski and C.-J.Seger. Asynchronous Circuits. Springer-Verlag. 1995.
2. Myers, C. Asynchronous Circuit Design. John Wiley & Sons, 2001.
3. S. Hauck, "Asynchronous design methodologies: An overview", Proc. IEEE, Vol. 83, No. 1, Jan. 1995, pp. 69-93.
4. S.H.Unger. Asynchronous Sequential Switching Circuits. Dep. of Electrical Engineering, Columbia University, Wiley Interscience, 1969.

5. L. A. Hollaar. Direct Implementation of Asynchronous Control Units. IEEE Transactions on Computers, Vol. C-31, No. 12, December 1982.

6. C. P. Sotiriou. Design of an Asynchronous Processor. PhD thesis, Institute for Computing Systems Architecture, Division of Informatics, University of Edinburgh, 2001.

7. Андрюхин А.И. Алгоритмы параллельного логического моделирования и псевдослучайной генерации тестов для МОП-структур // Микроэлектроника. -1995, N 5. -с. 331-336.

8. Андрюхин А.И. Параллельное логическое моделирование МОП-структур на переключательном уровне. // Электронное моделирование. - 1996, N 2, -с. 88-92.

9. Андрюхин А.И. Параллельное многозначное логическое моделирование исправных и неисправных псевдобулевых схем. // Электронное моделирование -1997, N 1. -с. 58-63.

10. Андрюхин А.И., Сперанский Д.В. Иерархическая компилятивная система моделирования и генерации тестов // Техническая диагностика и неразрушающий контроль. -1994. N 2. -с. 71-78.

11. Андрюхин А.И. Параллельное моделирование неисправностей МОП-структур // Научные труды Донецкого государственного технического университета. Серия: Проблемы моделирования и автоматизации проектирования динамических систем. Выпуск 29. 2001 г., С.205-211.