

ИСПОЛЬЗОВАНИЕ VHDL ОПИСАНИЯ ПРИ РЕАЛИЗАЦИИ БЛОКА КОДИРОВАНИЯ ДАННЫХ СИСТЕМЫ РАДИОЧАСТОТНОЙ ИДЕНТИФИКАЦИИ

В. С. Зайцев, В. Я. Степанец

*Белорусский государственный университет, кафедра
математической кибернетики
Минск, Беларусь
zaicevvs@tut.by, stepanets@bsu.by*

Приведен опыт разработки блока кодирования данных микросхемы радиочастотной идентификации с протоколом обмена ISO-18000-6С. Микросхема предназначена для использования в приложениях с рабочей частотой 860-960МГц.

Ключевые слова: VHDL, РЧИД-система, граф переходов, функция выхода, код Миллера, код FM0, автоматизированный синтез схем.

БЛОК КОДИРОВАНИЯ ДАННЫХ РЧИД-МИКРОСХЕМЫ С ПРОТОКОЛОМ ОБМЕНА ISO18000-6С

Проектирование Систем-на-Кристалле предусматривает применение макроблоков различного типа [1–3] и языков VHDL, VERILOG [4]. При этом последние используются не только для описания поведения создаваемого изделия (макроблока), его моделирования, но и для синтеза реализующей его логической схемы [5, 6]. Поэтому эффективность их применения определяется хорошим знанием их синтаксиса, семантики и возможностей средств синтеза [7, 8]. Далее описывается опыт использования языка VHDL при проектировании блока цифровой части микросхемы радиочастотной идентификации (РЧИД) [9, 10]. Одной из основных функций цифровой части системы РЧИД является кодирование информации, пересылаемой между ее передатчиком и приемником. Выбор типа кодирования во многом определяет надежность и помехозащищенность передачи данных. Наиболее часто используемые форматы кодирования данных в системах РЧИД – это кодирование с использованием кодов Миллера, FM0, Манчестер [11–13]. В настоящий момент производители такого рода систем включают поддержку в одном устройстве кодировок Миллера и FM0, либо всех трех указанных выше форматов кодирования одновременно.

К РЧИД системам, поддерживающим две кодировки, относятся системы, работающие в соответствии с международным стандартом ISO 18000-6С. Архитектура блока кодирования такой системы представлена на рис. 1.

Опишем алгоритм получения кода Миллера с помощью графа переходов и зададим значения функция выхода (рис. 2).

Блок цифровой обработки

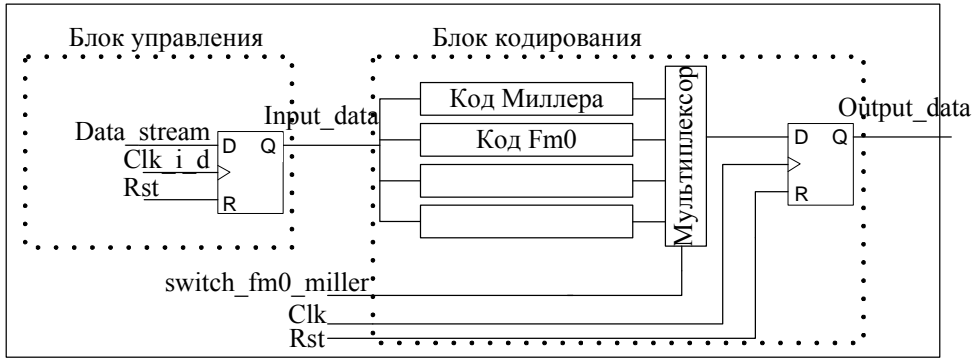


Рис.1. Архитектура блока кодирования РЧИД-системы

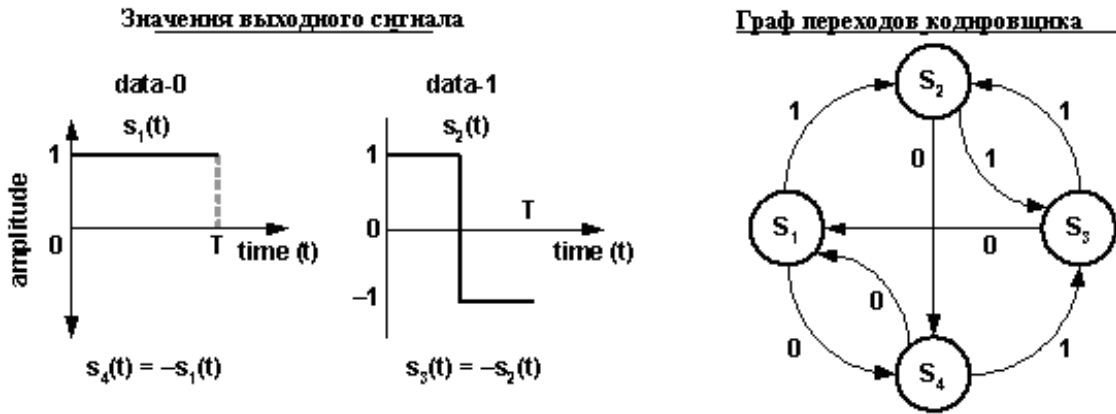


Рис.2. Граф переходов блока, формирующего код Миллера

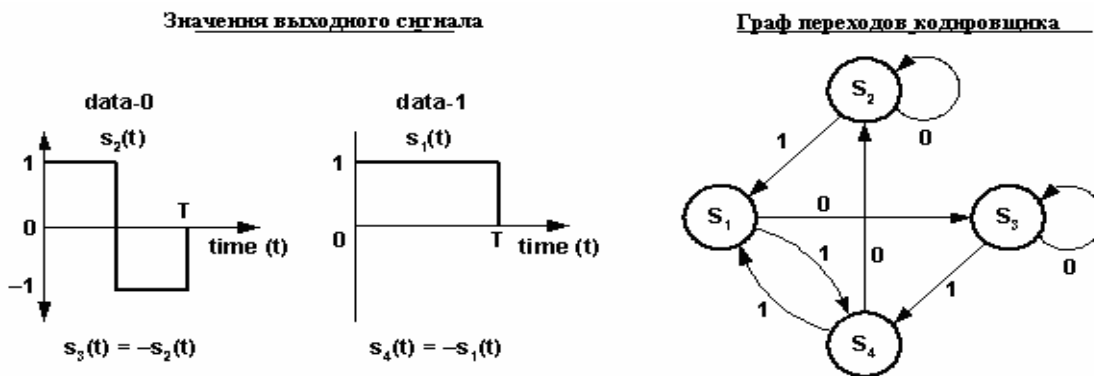


Рис.3. Граф переходов блока, формирующего код FM0

На рисунке 3 приведен граф переходов устройства, выполняющего FM0 кодирование.

Обратим внимание, что графы переходов схожи по структуре. Это потенциально позволяет объединить блоки кодирования таким образом, чтобы результат объединения занимал меньшую площадь, чем исходная схема (рисунок 4). Покажем, что такое объединение возможно.

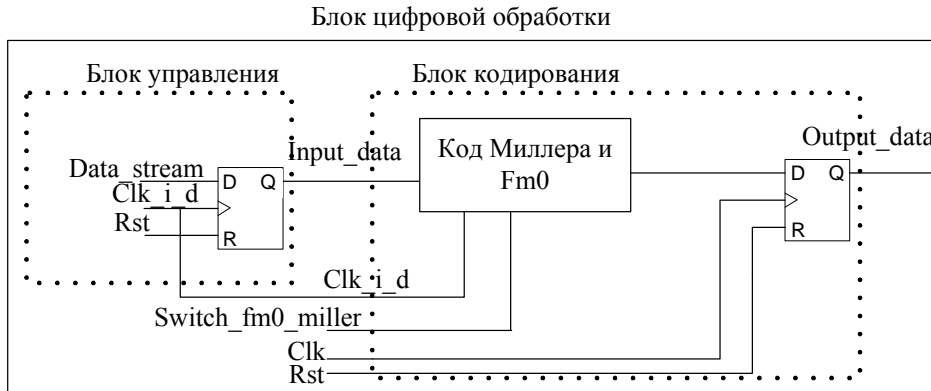


Рис.4. Архитектура блока кодирования после объединения кодировщиков

Опишем функцию перехода и функцию выхода для графа кодирования Миллера и введем замену переменных (1), получим новые функции перехода и выхода (2). Сравним полученные функции с функциями перехода и выхода для кодирования FM0. Можно заметить, что из одной функции перехода можно получить другую, изменив два условия перехода, а путем замены переменных получить соответствующую функцию выхода (таблица 1).

$$\begin{aligned}
 S1 &= a(S4,0), S1 = a(S3,0) & z = z(S1) &= 1 & s1' &= s3 \\
 S2 &= a(S1,1), S2 = a(S3,1) & z = z(S4) &= 0 & s2' &= s4 \\
 S3 &= a(S2,1), S3 = a(S4,1) & z = z(S2) &= \text{data_clk} & s3' &= s1 \\
 S4 &= a(S1,0), S4 = a(S2,0) & z = z(S3) &= \text{not data_clk} & s4' &= s2
 \end{aligned} \tag{1}$$

$$\begin{aligned}
 S3' &= a(S2,0), S3' = a(S1,0) & S1' &= a(S4,1), S1' = a(S2,1) & z = z(S1) &= \text{not data_clk} \\
 S4' &= a(S3,1), S4' = a(S1,1) & S2' &= a(S3,0), S2' = a(S4,0) & z = z(S2) &= 0 \\
 S1' &= a(S4,1), S1' = a(S2,1) & S3' &= a(S2,0), S3' = a(S1,0) & z = z(S3) &= 1 \\
 S2' &= a(S3,0), S2' = a(S4,0) & S4' &= a(S3,1), S4' = a(S1,1) & z = z(S4) &= \text{data_clk}
 \end{aligned} \tag{2}$$

Данное преобразование было описано на языке VHDL (рис. 5) и использовано для выполнения автоматического синтеза совмещенного блока кодирования. При этом были проверены три различных варианта явного представления конечного автомата на языке VHDL:

- все описание было размещено в одном операторе process;
- комбинационная часть функций перехода и выхода, а также синхронная часть преобразования были описаны в отдельных операторах process;
- комбинационная часть была описана конструкцией when - else, а синхронная – с использованием оператора process.

При этом для каждого из таких описаний были проверены установки синтезатора в части выбора способа кодирования состояний: auto, binary, gray.

Сравнение функции перехода и выхода

Миллер	FM0
$S1' = a(S4', 1)$	$S1 = a(S4, 1)$
$S1' = a(S2', 1)$	$S1 = a(S2, 1)$
$S2' = a(S3', 0)$	$S2 = a(S2, 0)$
$S2' = a(S4', 0)$	$S2 = a(S4, 0)$
$S3' = a(S2', 0)$	$S3 = a(S3, 0)$
$S3' = a(S1', 0)$	$S3 = a(S1, 0)$
$S4' = a(S3', 1)$	$S4 = a(S3, 1)$
$S4' = a(S1', 1)$	$S4 = a(S1, 1)$
$z = z(S1') = \text{not data_clk}$	$z = z(S1) = 1$
$z = z(S2') = 0$	$z = z(S2) = \text{data_clk}$
$z = z(S3') = 1$	$z = z(S3) = \text{not data_clk}$
$z = z(S4') = \text{data_clk}$	$z = z(S4) = 0$

```

begin
    input_data_i <= input_data;
coder_proc: process (clk_i_d, rst)
begin -- process coder_proc
    if rst = '1' then -- asynchronous reset (active low)
        state <= a1;
    elsif clk_i_d'event and clk_i_d = '1' then -- rising clock edge
        state <= statei;
    end if;
end process coder_proc;
change <= a1 when switch_fm0_miller = '1' else -- функция перехода
    a4;
statei <=
    a1 when input_data_i = '0' and (state = change or state = a3) else
    a2 when input_data_i = '1' and (state = a1 or state = a3) else
    a3 when input_data_i = '1' and (state = a2 or state = a4) else
    a4;
STATEjk <= a1 when STATE = a3 else -- замена переменных
    a2 when STATE = a4 else
    a3 when STATE = a1 else
    a4;
STATEj <= STATEjk when switch_fm0_miller = '1' else -- замена
    STATE;
output_datai <= -- функция выхода
    '1' when STATEj = a1 else
    clk_i_d when STATEj = a2 else
    not clk_i_d when statej = a3 else
    '0';
end beh2;

```

Рис.5. VHDL-описание совмещенного блока кодирования

Сравнение результатов синтеза, полученных при использовании трех указанных выше VHDL текстов, показало, что стиль описания не имеет принципиального значения при использовании системы синтеза Leonardo Spectrum фирмы MentorGraphics. Поэтому способ описания, который позволяет сделать VHDL код наиболее читабельным и воспроизводимым, можно считать наиболее приемлемым (рис. 5). При этом для получения наилучшего результата в качестве способа кодирования состояний следует использовать установку gray.

Время обработки в разработанном блоке не зависит от способа кодировки, а площадь реализации меньше суммарной площади исходных блоков, выполняющих отдельные кодировки (площадь реализации кода Миллера была 1330 мкм², кода FM0 -1327 мкм², а площадь разработанного блока - 1709 мкм²). В ходе разработки использована библиотека с технологическими нормами 0,35 мкм.

Описанные выше исследования являлись составной частью разработки на ОАО «ИНТЕГРАЛ» кристалла микропроцессора, предназначенного для использования в системе РЧИД, работающей на частоте 850 – 960 МГц по протоколу ISO 18000-6 часть С EPC Gen2. Дальнейшие исследования предусматривают разработку параметризуемого VHDL описания макроса цифровой обработки для использования при создании различных РЧИД систем.

ЛИТЕРАТУРА

1. Казеннов, Г. Г. Основы проектирования интегральных схем и систем / Г. Г. Казеннов. М.: БИНОМ. Лаборатория знаний, 2005. 295 с.
2. Суворова, Е. А. Проектирование цифровых систем на VHDL / Е. А. Суворова, Ю. Е. Шейнин. СПб.: БХВ-Петербург, 2003. 576 с.
3. Степанец, В. Я. О возможности решения проблемы синтеза полужестких макросов для проектирования Систем-на-Кристалле / В. Я. Степанец // Дискретная математика, алгебра и их приложения (ДИМА 09): тезисы докладов Междунар. науч. конф., Минск, 2009. Минск, 2009. С. 112–114.
4. Поляков, А. К. Языки VHDL и VERILOG в проектировании цифровой аппаратуры / А. К. Поляков. М.: СОЛОН-Пресс, 2003. 320 с.
5. Бибило, П. Н. Синтез логических схем с использованием языка VHDL / П. Н. Бибило. М.: СОЛОН-Р, 2002. 268 с.
6. Бибило, П. Н. Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum. / П. Н. Бибило. М.: СОЛОН-Пресс, 2005. 384 с.
7. Stepanets, V. Development of E-Learning Tools for Training of Specialists for Electronics at the BSU in Frame of RESON-project / V. Stepanets, E. Krylova // Proceeding of 49. Internationales Wissenschaftliches Kollquium (49. IWK) 27–30.09.2004, Ilmenau, Vol. 2 P. 492–497.
8. Бибило, П. Н. VHDL. Эффективное использование при проектировании цифровых систем / П. Н. Бибило, Н. А. Авдеев. М.: СОЛОН-Пресс, 2006. 344 с.
9. Авдеев, Н. А. Микросхема радиочастотной идентификации с протоколом обмена ISO-15693 / Н. А. Авдеев, А. И. Белоус, В. С. Зайцев, А. И. Колб // Электронные компоненты. 2010. № 8. С. 75–78.
10. Белоус, А. И. Микросхема радиочастотной идентификации IZ2805 / А. И. Белоус, В. С. Зайцев, А. И. Колб // Электронные компоненты. 2011. № 6. С. 53–56.
11. Зайцев, В. С. VHDL описание и реализация блока кодирования системы радиочастотной идентификации. / В. С. Зайцев, В. Я. Степанец // Материалы VI Междунар. конф. «Информационные системы и технологии (IST'2010)», Минск (Беларусь), 24–25 ноября 2010 г. С. 565–568.
12. Zaitsau, Viktor. The Expeirence of Use of VHDL Synthesis in Designing Radio Frequency Identification System / Viktor Zaitsau, Vladimir Stepanets // CADSM'2011, 23–25 Febrary, 2011, Polyana-Svalyava (Zakarpatitya), UKRAINE. P. 136–141.
13. (IZ2817) Микросхема интегральная-транспондер со встроенной памятью и функциями чтения/записи с поддержкой системы защиты данных. Свидетельство РФ о государственной регистрации топологии интегральной схемы № 2010630053 от 14 апреля 2010 г. / И. С. Скоковская, В. С. Зайцев,

Ю. А. Макаревич, А. И. Колб, В. С. Сякерский. Правообладатель: Производственное республиканское унитарное предприятие «Завод полупроводниковых приборов» (ВУ).
