

4 **Филиппов, В. М.** Управление техническим состоянием верхнего строения пути : автореф. дис. ... д-ра техн. наук. – М., 1997 [Электронный ресурс]. – Режим доступа : <http://tekhnosfera.com/upravlenie-tehnicheskim-sostoyaniem-verhnego-stroeniya-puti>. – Дата доступа : 23.09.2021.

5 **Марков, А. А.** Актуальные проблемы дефектоскопии рельсов и пути их решения / А. А. Марков [Электронный ресурс]. – Режим доступа : http://www.radioavionica.ru/activities/sistemy-nerazrushayushchego-kontrolya/articles/files/razrab/st_2016_1.pdf. – Дата доступа : 23.09.2021.

6 **Lancaster, Gavin.** DAS: Готовое решение для ж/д отрасли / Gavin Lancaster // Ultimate Rai. – 2018. – № 2. – С. 23–25.

7 Сходы из-за изломов рельса [Электронный ресурс]. – Режим доступа : https://vuzlit.ru/1001608/shody_izlomov_relsa. – Дата доступа : 23.09.2021.

8 **Плотников, О.** Практический опыт эксплуатации сети LoRaWAN. Заметки IoT-провайдера / О. Плотников [Электронный ресурс]. – Режим доступа : <http://nag.ru/go/text/101403/>. – Дата доступа : 23.09.2021.

9 УИС. Доклад Марка Энтони на IV Форуме по датчикам колес, Вена, 4–6 октября 2017 г.

УДК 004.052.42

СИНТЕЗ УСТРОЙСТВ УПРАВЛЕНИЯ СТРЕЛОЧНЫМ ПЕРЕВОДОМ НА ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ ИНТЕГРАЛЬНЫХ СХЕМАХ

А. В. ПАШУКОВ

Российский университет транспорта (МИИТ), г. Москва

Программируемые логические интегральные схемы (ПЛИС), несмотря на высокую конкуренцию, смогли найти применение во многих отраслях промышленности. Так, ПЛИС применяются в атомной промышленности [1, 2], авиации, военной технике [3], на железных дорогах [4, 5] и т. д.

Использование ПЛИС в области СЦБ на железнодорожном транспорте имеет ряд преимуществ. Среди них выделяются:

- существенное изменение себестоимости создания проекта;
- возможность реализации сложных параллельных алгоритмов;
- высокое быстродействие, необходимое для диагностики устройств ЖАТ;
- возможность программирования или реконфигурации непосредственно в готовой системе;
- аппаратная реализация алгоритмов работы систем ЖАТ;
- уменьшение занимаемого объема необходимой аппаратуры по сравнению с релейными схемами;
- уменьшение затрат электроэнергии.

В данной работе рассматривается возможность создания на ПЛИС схем управления объектами напольной автоматизации железнодорожного транспорта на примере схемы управления стрелочным переводом.

При синтезе схемы управления стрелочным переводом нужно описать логику ее работы в виде конечного автомата, который будет являться основой схемы управления. Для этого воспользуемся графом переходов (рисунок 1).

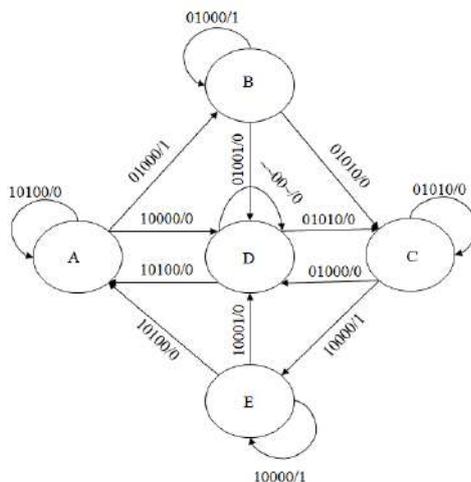


Рисунок 1 – Граф переходов схемы управления стрелочным переводом:
А – плюсовое положение стрелки; В – потеря контроля при переводе «←»; С – минусовое положение стрелки;
Е – потеря контроля при переводе стрелки в «+»

Рассмотрим наиболее простой частный случай синтеза схемы, где используются только 5 состояний: «Плюсовое положение стрелки», «Минусовое положение стрелки», «Потеря контроля при переводе в плюсовое положение», «Потеря контроля при переводе в минусовое положение» и «Защитное состояние». В дальнейшем добавятся состояния, соответствующие замкнутости в маршруте, передаче на макет и т. д.

В начальном состоянии стрелка находится в одном из двух состояний: «Плюсовое положение» или «Минусовое положение». При потере контроля положения стрелки без сигнала на перевод схема перейдет в защитное состояние, где будет находиться до устранения неисправности. При получении команды на перевод стрелка перейдет в состояние «Потеря контроля при переводе» и будет в нем оставаться до получения контроля крайнего положения стрелки или до истечения времени перевода стрелки. Если перевод завершен успешно, то стрелка примет одно из начальных состояний. Если время перевода истекло, а контроля положения нет, то схема перейдет в защитное состояние.

При синтезе конечного автомата для предотвращения опасных отказов стоит учитывать возможные неисправности и способы их обнаружения. Более подробно про неисправности и способы их обнаружения описаны в [6]. В данной работе для обнаружения неисправностей в спроектированной схеме были применены модульные коды с суммированием [7, 8]. Для этого заданные состояния схемы кодируются с помощью указанных кодов.

Синтез конечного автомата был осуществлен на ПЛИС фирмы Altera, семейства MAX II с помощью программной среды Quartus Prime. Блок логики переходов между состояниями представлен ниже:

```
case (state)
A:  if ( i==1 and i2==0 and i3==0 and i4==0 and i5=0) next_state = D;
    if ( i==0 and i2==1 and i3==0 and i4==0 and i5=0) next_state = B;
    else next_state = A;
B:  if ( i==0 and i2==1 and i3==0 and i4==1 and i5=0) next_state=C;
    if ( i==0 and i2==1 and i3==0 and i4==0 and i5=1) next_state=D;
    else next_state=B;
C:  if ( i==0 and i2==1 and i3==0 and i4==0 and i5=0) next_state=D;
    if ( i==1 and i2==0 and i3==0 and i4==0 and i5=0) next_state=E;
    else next_state=C;
D:  if ( i==1 and i2==0 and i3==1 and i4==0 and i5=0) next_state=A;
    if ( i==0 and i2==1 and i3==0 and i4==1 and i5=0) next_state=C;
    else next_state=D;
E:  if ( i==1 and i2==0 and i3==1 and i4==0 and i5=0) next_state=A;
    if ( i==1 and i2==0 and i3==0 and i4==0 and i5=1) next_state=D;
    else next_state=E;
endcase,
```

где $i_1, i_2, i_3, i_4, i_5, 0$ – входные и выходные данные; i_1 – команда на перевод в плюсовое положение; i_2 – команда на перевод в минусовое положение; i_3 – контроль плюсового положения; i_4 – контроль минусового положения; i_5 – таймер перевода стрелки; 0 – состояние двигателя электропривода (вкл./выкл.).

Стоит отметить, что применение модульных кодов не приводит к резкому увеличению избыточности, что позволяет их эффективно применять не только для устройств, реализованных на ПЛИС, но и для любых типов устройств, имеющих в основе элементы программируемой логики.

Предложенный вариант синтеза схем управления на железнодорожном транспорте дает возможность создания новых систем обеспечения движения поездов на высокоскоростных магистралях, где предъявляются повышенные требования к быстрдействию, надежности и безопасности.

Список литературы

- 1 Проект цифровой управляющей системы безопасности для энергоблоков АЭС с реакторами ВВЭР / А. В. Кудрявцев [и др.]. – М. : Национальный исследовательский ядерный университет «МИФИ», 2012. – 144 с.
- 2 New control systems for emergency protection of power reactors / A. A. Zaikin [et al.] // Ядерные измерительно-информационные технологии. – 2005. – № 3. – С. 30–55.
- 3 Перспективы использования FPGA-технологий в системах железнодорожной автоматики / М. Л. Малиновский [и др.] // Вісник Харківського національного технічного університету сільського господарства імені Петра Василенка. Технічні науки. – Харьков, 2012, Вип. 130: Проблеми енергозабезпечення та енергозбереження в АПК України. – С. 79–81.

4 Кабецкий, А. Г. Методы и инструментальные средства построения логических устройств электрической централизации на базе программируемых логических интегральных схем / А. Г. Кабецкий, Д. С. Марков // Известия Петербургского университета путей сообщения. – 2010. – № 2. – С. 168–173.

5 Dobiáš, R. FPGA Based Design of Railway's Interlocking Equipment / R. Dobiáš, H. Kubátová // Proceedings of EUROMICRO Symposium on Digital System Design. – 2004. – P. 467–473.

6 Сапожников, Вл. В. Синтез систем управления движением поездов на железнодорожных станциях с исключением опасных отказов / Вл. В. Сапожников. – М. : Наука, 2021. – 229 с.

7 Сапожников, В. В. Коды с суммированием для систем технического диагностирования. Т. 1. Классические коды Бергера и их модификации : [монография] / В. В. Сапожников, Вл. В. Сапожников, Д. В. Ефанов. – М. : Наука, 2020. – 383 с.

8 Сапожников, В. В. Коды с суммированием для систем технического диагностирования. Т. 2. Взвешенные коды с суммированием : [монография] / В. В. Сапожников, Вл. В. Сапожников, Д. В. Ефанов. – М. : Наука, 2021. – 455 с.

УДК 656.25

ОЦЕНКА НЕЗАВИСИМОСТИ ОТКАЗОВ В ДИВЕРСИТЕТНОМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ С ПОМОЩЬЮ EL-МОДЕЛИ

С. Н. ХАРЛАП, А. Ю. КУЛАЖЕНКО

Белорусский государственный университет транспорта, г. Гомель

Современные микроэлектронные системы железнодорожной автоматики реализуются как многоканальные системы. В таких системах технологические алгоритмы, связанные с функциональной безопасностью, выполняются параллельно в различных каналах с последующим сравнением результатов работы. Управляющие воздействия вырабатываются только в том случае, если каналы выдали одинаковые результаты. Безопасность таких систем базируется на предположении, что любой отказ или ошибка в программном обеспечении приведут к различной реакции каналов на одни и те же входные воздействия, что будет обнаружено схемой сравнения и переведет систему в защитное состояние.

Такое поведение системы при отказах или ошибках в программном обеспечении можно обеспечить только в случае независимости возникновения отказов и проявления ошибок программного обеспечения в разных каналах. В соответствии с МЭК61508 основным способом обеспечения независимости отказов в различных каналах многоканальной системы является диверситет (разнообразие). При этом диверситет может быть функциональным, основанным на различных способах решения одной и той же задачи, и диверситетом технологий.

К сожалению, в стандарте не даются методы оценки достигнутого диверситета, что приводит к убежденности, что любой диверситет обеспечивает необходимую независимость отказов. В то же время имеются публикации, показывающие ложность данного утверждения, в частности предложенная в конце прошлого века формализованная модель (EL-модель) позволяет получить вероятностную оценку диверситетных программ и математически доказывает, что невозможно достичь независимости отказов в различных версиях программы, даже если версии разработаны независимо в соответствии с действующими стандартами.

Рассмотрим в качестве примера оценку независимости отказов управляющей программы, выполняющей вычисление функции алгебры логики $Y=A \cdot (B+\bar{C}+D \cdot A \cdot \bar{D})+D \cdot \bar{A} \cdot (B+\bar{B})$. Функция реализована следующими способами:

π_1 – непосредственное вычисление ФАЛ;

π_2 – минимизация (преобразование) ФАЛ и ее непосредственное вычисление. Функция после минимизации: $Y=A \cdot (B+\bar{C})+D \cdot \bar{A}$;

π_3 – вычисление методом бинарных программ;

π_4 – вычисление по таблице истинности (метод адресных переходов).

Методы π_1 и π_2 реализованы непосредственным вычислением ФАЛ с помощью логических операций AND, OR, NOT. Метод π_3 представляет собой вычисление ФАЛ с помощью проверки ряда условий, например, если $A = 0$ и $D = 0$, то $Y = 0$. Метод π_4 основан на предварительном вычислении таблицы истинности функции и записи ее в определенную область памяти. Алгоритм предусматривает вычисление по значениям входных переменных адреса ячейки таблицы истинности, в которой хранится результат. Методы π_1 , π_3 и π_4 считаются взаимно диверситетными. Для методов π_1 и π_2 взаимный диверситет считается недостаточным.