Применение адаптивного сигнатурного анализа для контроля потока управления микроконтроллерных IP-ядер

Мусин[1] С.Б.; Базылев[1] Е.Н.; Иванюк[2] А.А.
[1]ПОИТ ФКСиС
[2]ВМиП ФИТиУ
БГУИР
Минск, Беларусь
e-mail: musin@bsuir.by

***Аннотация* — Предлагается архитектура адаптивного сигнатурного анализатора для контроля памяти программ микроконтроллеров встроенных систем, рассмотрен прототип устройства контроля на базе IP-ядра микроконтроллеров семейства PIC16.**

Ключевые слова: контроль потока управления, сигнатурный анализ

I. Введение

В настоящее время разработчики встроенных систем критических приложений все чаще отдают предпочтение коммерческим микроконтроллерным IP-ядрам. Производители данных устройств не используют высоконадёжную элементную базу, но предлагают хорошую скорость работы и потребление энергии при низкой стоимости [1]. Сложность применения коммерческих микросхем в критических приложениях заключается в том, что их надёжность и качество зачастую являются недостаточными. Поэтому одним из важнейших аспектов обеспечения надежного функционирования встроенных систем критических приложений является использование методов контроля для защиты от ошибок.

Наличие ошибок в потоке управления может привести к повреждениям данных, завершению работы процессов, неправильной работе программы без выдачи сообщений об ошибках. Согласно исследованиям приведенным в [2] треть всех ошибок приложений не работающих интенсивно с данными приходится на ошибки в потоке управления.

II. Контроль потока управления

В основе методов контроля потока управления [3] лежит идея разделения программы на блоки, которые содержат только последовательные команды, без ветвлений. Каждая команда перехода формирует новый блок. При компиляции программы для каждого блока рассчитывается и сохраняется эталонная сигнатура, которая отражает корректную последовательность выполнения команд. Во время выполнения программы сторожевой таймер рассчитывает рабочую сигнатуру. Если возникает сбой, последовательность команд программы изменяется, что приводит к изменению вновь рассчитанной рабочей сигнатуры. Факт различия эталонной и рабочей сигнатур говорит о том, что возникла ошибка в потоке выполнения программы.

Несмотря на большое количество публикаций по теме контроля потока управления, исследования в данной области по-прежнему актуальны [4]. Открытой остается проблема снижения аппаратных затрат на схему контроля, увеличение кратности обнаруживаемых ошибок, уменьшение латентности ошибки.

III. Адаптивный сигнатурный анализ

Представим последовательный участок программы в виде вектора пар: ({*адрес*, *команда*},... ,{*адрес*, *команда*})

Для получения сигнатуры S на сигнатурном анализаторе сжимаются вектора ассоциированные с каждой i-й парой {*адрес*, *команда*}:

$$S = \oplus\sum_i s_i = \oplus\sum_i \{SA^i,\ \ SC^i\}.$$

Где значение s_i сформировано из префиксной части – SA^i и суффиксной части – SC^i, которые являются компактными характеристиками адресов и самих команд программы соответственно.

Пусть размер префиксной части составляет 13 бит (размерность шины адреса), размер суффиксной – 4 бита, при размере слова памяти программ – 14 бит. Тогда биты суффиксной части SC^i будут определяться исходя из битов b_j каждой i-й команды следующим образом:

$$\begin{cases} SC_0^i &= b_1^{\,i} \oplus b_3^{\,i} \oplus b_5^{\,i} \oplus b_7^{\,i} \oplus b_9^{\,i} \oplus b_{11}^{\,i} \oplus b_{13}^{\,i}, \\ SC_1^i &= b_2^{\,i} \oplus b_3^{\,i} \oplus b_6^{\,i} \oplus b_7^{\,i} \oplus b_{10}^{\,i} \oplus b_{11}^{\,i}, \\ SC_2^i &= b_4^{\,i} \oplus b_5^{\,i} \oplus b_6^{\,i} \oplus b_7^{\,i} \oplus b_{12}^{\,i} \oplus b_{13}^{\,i}, \\ SC_3^i &= b_8^{\,i} \oplus b_9^{\,i} \oplus b_{10}^{\,i} \oplus b_{11}^{\,i} \oplus b_{12}^{\,i} \oplus b_{13}^{\,i}. \end{cases}$$

При этом префиксная часть SA^i устанавливается равной адресу команды a^i, в случае если общая четность битов команды положительна и нулю в противном случае:

$$SA^i = a_i \times \left(\oplus\sum_{j=0}^{13} b_j^{\,i} \right).$$

Достоверность такого способа применения адаптивного сигнатурного анализа для контроля потока управления соответствует достоверности помехоустойчивого кода Хемминга. Коррекция Для повышения обнаруживающей и корректирующей способности следует использовать схему сжатия адресов построенную на базе дуального кода Рида–Маллера [5].

IV. Реализация

Сигнатурный анализатор (рис.1) располагается между ядром микроконтроллера и памятью программ.

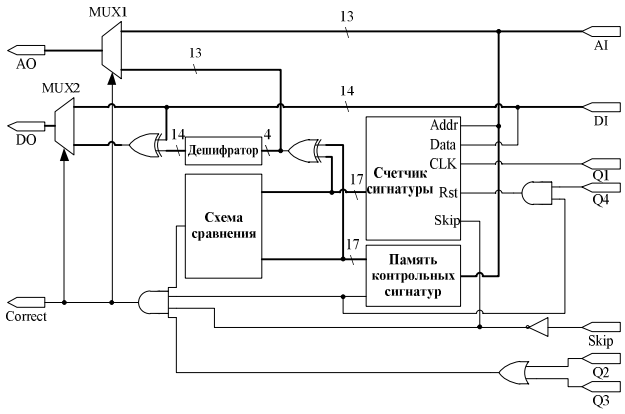

Рис. 1. Схема сигнатурного анализатора

При отсутствии ошибок мультиплексоры MUX1 и MUX2 свободно пропускают на выход адрес с шины AI и команду с шины DI от IP-ядра.

Цикл выполнения команды делится на четыре такта, которые задаются сигналами Q1, Q2, Q3 и Q4. Микроконтроллер имеет двухуровневый конвейер: при выполнении текущей команды на шину адреса подаётся адрес следующей ячейки памяти, во время последнего такта Q4 происходит загрузка регистра команд. Счетчик сигнатуры подсчитывает сигнатуру при получении сигнала Q1 и сбрасывается сигналом от памяти контрольных сигнатур в течение действия сигнала Q4.

При выполнении команд переходов происходит приостановка конвейера и генерируется сигнал Skip, при получении которого сигнатурный анализатор не выполняет расчет сигнатуры. Расчет сигнатур производится производится параллельно с работой IP-ядра схемой представленой на рис. 2.

Схема сравнения проверяет на равенство эталонную и рабочую сигнатуры. 4 младших бита суммы по модулю два сигнатур поступают на вход дешифратора, который генерирует вектор ошибки. При не совпадении сигнатур, в течение действия сигналов Q2 и Q3 генерируется сигнал коррекции Correct, который переключает мультиплексоры, и на шину адреса AO подаётся адрес корректируемой ячейки, а на шину данных DO подаётся скорректированное значение команды. Во время действия сигнала Q2 происходит чтение корректируемой ячейки, это значение складывается по модулю два с вектором ошибки от дешифратора, и полученное скорректированное значение записывается в память во время действия сигнала Q3.

VHDL-модуль для хранения контрольных сигнатур представляет собой ПЗУ, размер каждой ячейки которого равен 18 бит. Старший бит указывает, соответствует ли значению шины адреса значение контрольной сигнатуры или нет, соответственно генерируется флаг signFlag.

В качестве памяти программ использована блочная память (Block RAM), которая позволяет производить начальную инициализацию, а так же операции синхронного чтения и записи.

Разработана утилита на языке C++ для распаковки HEX-файла и генерации VHDL-описания модуля хранения контрольных сигнатур и модуля инициализации памяти программ.

В качестве опытного микроконтроллерного IP-ядра использовано 14-битное IP-ядро микроконтроллера PIC16 фирмы Microchip [6]. Схема контроля реализована на языке VHDL и синтезирована на ПЛИС Xilinx Spartan 2E с использованием системной платы Digilent D2-SB [7].

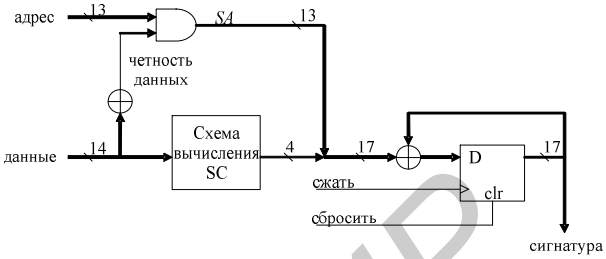

Рис. 2. Схема вычисления сигнатуры

Аппаратурные затраты на схему контроля составляют 17 D-триггеров, 1 компаратор, 27 мультиплексоров, 1 комбинационный сдвиговый регистр и 8 вентилей XOR.

V. Заключение

Разработанное экспериментальное устройство, обладает малыми аппаратурными затратами, что позволяет утверждать о применимости метода адаптивного сигнатурного анализа для контроля потока управления микроконтроллерных IP-ядер. Дальнейшие исследования направлены на увеличение крастности обнаруживаемых ошибок и уменьшение латентости ошибки.

[1] А.П. Поливанов, “Методы увеличения времени функционирования КМОП СБИС запоминающих устройств в составе бортовой аппаратуры космических аппаратов” : 05.27.01 Дис. канд.техн.наук.-М. РГБ, 2005

[2] S. Bagchi, “Hierarchical Error Detection in a SIFT Environment”, Ph.D. Thesis, Advisor: R. K. Iyer, University of Illinois at Urbana-Champaign, December 2000.

[3] Y.-Y. Chen, “Concurrent detection of control flow errors by hybrid signature monitoring”, IEEE Transactions on Computers, vol 54, Oct. 2005, pp. 1298 – 1313, doi: 10.1109/TC.2005.158

[4] D. Gizopoulos, M. Psarakis, S. V. Adve, P. Ramachandran, S.K.S. Hari, D. Sorin, A. Meixner, A. Biswas, and X. Vera. "Architectures for Online Error Detection and Recovery in Multicore Processors." To appear in Design, Automation & Test in Europe (DATE), March 2011.

[5] А.А. Иванюк, С.Б. Мусин, В.Н. Ярмолик, Использование адаптивного сигнатурного анализа для обнаружения многократных ошибок ОЗУ // Микроэлектроника. - 2007. - №3 (36). - С. 246-253.

[6] Microchip [Электронный ресурс]. – Электронные данные. – Режим доступа: http://www.microchip.com/.

[7] Digilent [Электронный ресурс]. – Электронные данные. – Режим доступа: http://www.digilentinc.com/.