

УДК 681.3.06

СИСТЕМА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ КОДЕКОВ ПОМЕХОУСТОЙЧИВЫХ КОДОВ КРОТКОЙ ДЛИНЫ

А.Н. Мальчуков, А.Н. Осокин

Институт «Кибернетический центр» ТПУ

E-mail: jgs@tpu.ru

Рассмотрена система автоматизированного проектирования быстродействующих кодеков эффективных помехоустойчивых кодов. Система позволяет проектировать кодеки линейных блочных кодов короткой длины с минимальной избыточностью по задаваемым пользователем параметрам: длине информационного блока и желаемому количеству исправляемых независимых ошибок.

Введение

При проектировании систем промышленной автоматизации и сетей сбора информации с датчиков целесообразно использовать помехоустойчивые коды для повышения достоверности передаваемых данных. В таких системах возможны независимые и пачки ошибок. Также в некоторых случаях сбора информации с датчиков возникает ситуация, при которой пропускная способность канала ограничена малой величиной (например, передача данных из скважин по гидравлическому или электромагнитному каналам), что в свою очередь требует от помехоустойчивых кодов минимальной избыточности (длины кодового слова), достаточной для исправления заданного класса ошибок. Для исправления независимых ошибок различной кратности широко используются линейные блочные помехоустойчивые коды небольшой длины: Хемминга, Рида-Маллера, БЧХ-коды (коды Боуза-Чоудхури-Хоквенгема) и др. Для непосредственного выбора помехоустойчивого кода под конкретные условия требуются глубокие знания теории помехоустойчивого кодирования.

Процедуры кодирования и декодирования для этих кодов различны. Вместо использования разных кодов для различных условий можно использовать вариации БЧХ-кодов. Для построения БЧХ-кодов необходимо решать задачу выбора образующего полинома или использовать готовые решения, описанные в литературе [1, 2]. Однако, предлагаемое в них построение БЧХ-кодов ограничивает длину кодового слова значениями равными $2^p - 1$ (где p – любое натуральное число). Остальные длины кодовых слов получаются путём укорачивания более длинных кодовых слов за счёт уменьшения разрядности информационного блока при сохранении количества контрольных разрядов. Укорачивание классических БЧХ-кодов вводит лишнюю избыточность, которая не несёт в себе дополнительных корректирующих возможностей, а лишь замедляет скорость при последовательной передаче данных, либо делает невозможным применение такого кода при ограниченной скорости передачи.

В связи с вышеизложенным, актуальной задачей является разработка системы проектирования кодеков помехоустойчивых кодов, которая позволит проектировать быстродействующие кодеки, исходя из длины информационного блока и желаемой кор-

ректирующей способности кода без специальных знаний в области помехоустойчивого кодирования с минимально возможной избыточностью.

Структура системы проектирования кодеков

Система состоит из следующих взаимосвязанных модулей (рис. 1):

1. Выбор образующего полинома.
2. Формирование помехоустойчивого кода.
3. Проектирование кода.
4. Реализация на программируемых логических интегральных схемах (ПЛИС).

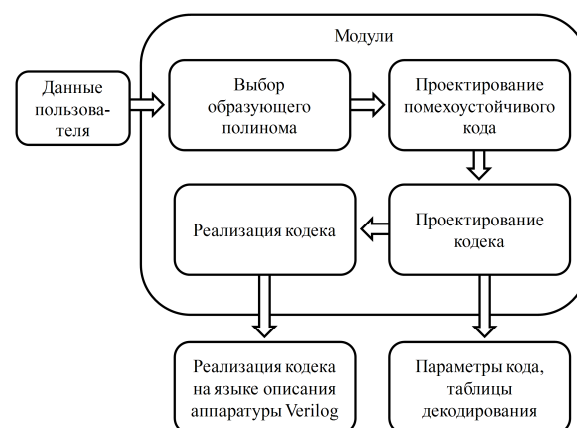


Рис. 1. Структура системы проектирования кодеков

В системе используются модифицированные циклические помехоустойчивые коды, в которых процедуры кодирования и декодирования основаны на делении кодового слова на образующий полином по модулю два. Основные отличия заключаются в выборе и предъявляемым требованиям к образующему полиному.

Данные пользователя

В качестве входных данных пользователь вводит длину информационного блока (количество бит данных, передаваемых в одном кодовом слове) и желаемую корректирующую способность кода (количество исправляемых кодом независимых ошибок).

Введённые пользователем два параметра поступают на вход модуля выбора образующего полинома.

Модуль выбора образующего полинома

На входе модуля — задаваемые пользователем параметры: разрядность информационной последовательности и желаемая корректирующая способность кода. Модуль работает с файлами данных, в которых содержатся заранее найденные и проверенные полиномы для различных вариаций пользовательских параметров.

В отличие от БЧХ-кодов, где к образующему полиному предъявляются требования (полином должен быть одним из множителей x^n-1 ; также он должен состоять из t множителей неприводимых полиномов), в разрабатываемой системе предъявляется единственное требование к искомому полиному — чтобы формируемый им помехоустойчивый код обладал заданной корректирующей способностью.

Для поиска образующих полиномов использовался авторский алгоритм:

1. Вычисляется минимальная длина контрольного блока (старшая степень образующего полинома) по формуле «граница Хэмминга» [1, 2]:

$$k = \left\lceil \log_2 \left(\sum_{i=0}^t C_n^i \right) \right\rceil,$$

где k — старшая степень образующего полинома, t — количество независимых ошибок, исправляемых кодом, n — разрядность кодового слова, $\lceil a \rceil$ — большее целое от a , C_n^i — число сочетаний i из n .

2. Выбирается полином из множества, состоящего из полиномов со старшей степенью равной k и весом, равным минимальному расстоянию кода d_m . Минимальное расстояние кода, исправляющего t независимых ошибок, вычисляется по формуле: $d_m = 2t + 1$ [1, 2].
3. Вычисляется минимальное кодовое расстояние кода (путём нахождения минимального веса среди всех кодовых слов, за исключением нулевого слова [1]), строящегося на основе выбранного полинома.
4. Если вычисленное кодовое расстояние удовлетворяет заданным требованиям, то переходим к п. 6, иначе переходим к п. 5.
5. Если проверены все полиномы из множества (множество задано в п. 2), увеличиваем k на единицу и переходим к п. 2.
6. Если все синдромы ошибок уникальны для построенного кода, то полином принимается в качестве образующего полинома и переходим к п. 7, иначе переходим к п. 5.
7. Проверяется способность кода исправлять пакеты ошибок длиной от $t+1$ до $n-1$.
8. Результат сравнивается с результатом предыдущего полинома, запоминается полином с лучшим результатом.
9. Если проверены все полиномы из множества (множество задано в п. 2), переходим к п. 10, иначе переходим к п. 2.
10. Конец.

После выбора и проверки образующие полиномы записываются в файл данных и в дальнейшем используются системой. Также необходимо отметить, что система позволяет легко изменять содержание файлов с образующими полиномами, если в будущем найдутся полиномы, на основе которых стоят коды с лучшими характеристиками.

Несмотря на то, что количество итераций алгоритма поиска было существенно сокращено, поиск образующих полиномов остаётся по-прежнему процессом ресурсоёмким. Верхнюю оценку количества итераций авторского алгоритма Q_a можно посчитать по формуле:

$$Q_a = C_{n-2}^{d-2} \cdot (2^m - 1) + 2^m.$$

Количество итераций Q при полном переборе вычисляется как:

$$Q = \sum_{i=3}^n (C_{n-2}^{i-2} \cdot (2^m - 1)) + 2^m.$$

Пусть $m=29$, $t=3$, тогда $n=47$ и количество итераций Q_a , которое потребуется авторскому алгоритму, равно $6,559 \cdot 10^{14}$. В случае полного перебора количество итераций Q будет равно $1,889 \cdot 10^{22}$.

Таблица. Длины кодовых слов: а) авторских кодов, б) БЧХ-кодов и в) длины пакетов ошибок

m/t	а				б				в			
	1	2	3	4	1	2	3	4	1	2	3	4
1	3	5	7	9	3	7	7	15	1	2	3	4
2	5	8	11	14	5	10	12	22	1	3	4	6
3	6	10	13	17	6	11	13	23	1	3	4	7
4	7	11	14	19	7	12	14	24	1	2	4	7
5	9	13	15	20	9	13	15	25	1	3	4	7
6	10	14	17	22	10	14	21	26	1	2	4	7
7	11	15	18	24	11	15	22	27	1	2	3	8
8	12	16	19	25	12	18	23	28	1	2	3	7
9	13	17	20	27	13	19	24	29	1	2	3	8
10	14	19	21	28	14	20	25	30	1	3	3	7
11	15	20	22	29	15	21	26	31	1	2	3	4
12	17	21	23	31	17	22	27	36	1	2	3	6
13	18	22	27	32	18	23	28	37	1	2	3	6
14	19	24	29	34	19	24	29	38	1	4	6	8
15	20	25	30	35	20	25	30	39	1	4	6	6
16	21	26	31	37	21	26	31	40	1	4	6	5
17	22	27	33	38	22	27	35	41	1	4	6	5
18	23	28	34	39	23	28	36	42	1	4	5	5
19	24	29	36	40	24	29	37	43	1	4	7	5
20	25	30	37	42	25	30	38	44	1	4	7	7
21	26	31	38	43	26	31	39	45	1	4	7	7
22	27	33	39	45	27	34	40	46	1	3	7	6
23	28	34	40	46	28	35	41	47	1	3	5	6
24	29	35	41	47	29	36	42	48	1	3	5	6
25	30	37	42	48	30	37	43	49	1	4	5	6
26	31	38	43	49	31	38	44	50	1	3	5	5
27	33	39	45	51	33	39	45	51	1	3	7	6
28	34	40	46	52	34	40	46	52	1	3	4	6
29	35	41	47	53	35	41	47	53	1	3	4	5

Результаты работы алгоритма на суперкомпьютерном кластере «СКИФ-политех» сведены в таблицу, графа а. Для сравнения в графе б показаны длины кодовых слов БЧХ-кода для тех же параметров. Значе-

ния в графе b вычислены с помощью программы bch3.c [3]. В таблице приведены длины пакетов ошибок, которые авторские коды могут исправлять в дополнение к исправлению t независимых ошибок.

В таблице m – разрядность информационного блока. Разница между длинами кодовых слов авторских и БЧХ-кодов отражена на графиках рис. 2 для $t=2-4$.

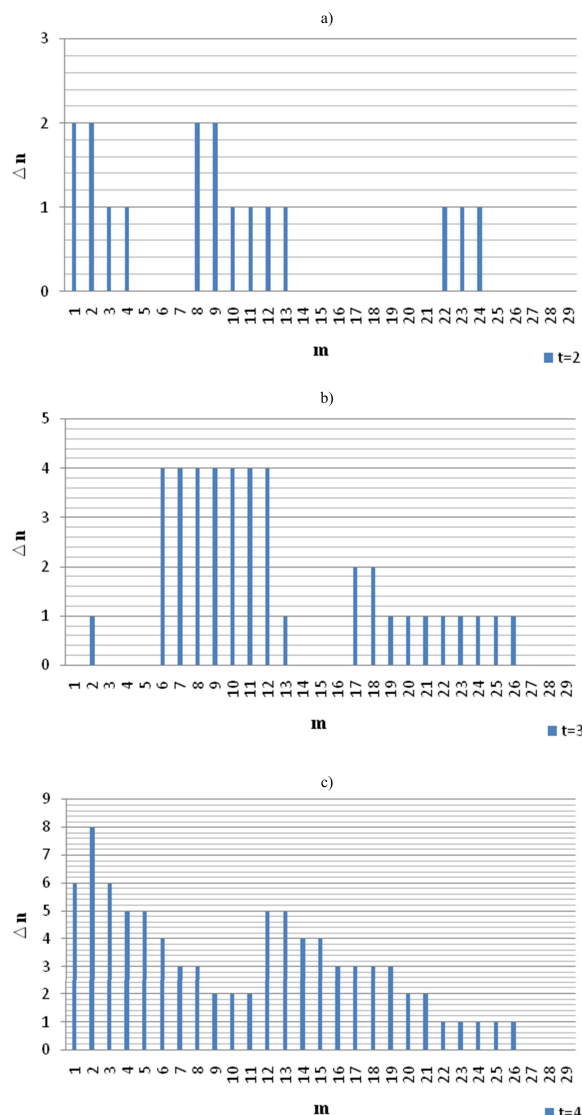


Рис. 2. Графики зависимостей разницы n (БЧХ и авторских кодов) от m для: а) $t=2$, б) $t=3$, в) $t=4$

В таблице на пересечении строк и столбцов указана длина пакета, которую код может исправлять в дополнение к исправлению t независимых ошибок. Если длина пакета больше t – это означает, что код исправляет пакетные ошибки длиной от $t+1$ до указанного числа, иначе пакетные ошибки не исправляются, а исправляются только t независимых ошибок. Это дополнительное свойство кода проверялось на найденных образующих полиномах, которые специально не искались с этим уклоном, в данный момент проводится дополнительный поиск полиномов без изменения разрядности кодо-

вых слов, но с прицелом на лучшие показатели в области исправления пакетов ошибок.

Из рис. 2 видно, что длины кодовых слов помехоустойчивых кодов (для $m=1-29$), построенные на основе авторских образующих полиномов, не превышают длины кодовых слов БЧХ-кодов, а в большинстве случаев длины авторских кодовых слов короче, чем у БЧХ-кодов – что является показателем лучшего результата.

Полученные результаты сопоставимы с результатами, опубликованными группой разработчиков программы MAGMA (Markus Grassl, Greg White, Damien Fisher) [4].

Матрица задержек для ПЛИС семейства Flex10k фирмы Altera представлена на рис. 4.

В итоге на выход модуля поступает выбранный образующий полином и вычисленная длина кодового слова, которые передаются в модуль формирования помехоустойчивого кода.

Модуль формирования помехоустойчивого кода

Основной операцией в процедурах кодирования и декодирования помехоустойчивых кодов системы является операция нахождения остатков (от деления кодового слова на образующий полином по модулю два). Для увеличения быстродействия декодеров, проектируемых системой, разработан авторский алгоритм матричного деления полиномов в арифметике по модулю два [5], позволяющий быстро вычислять контрольный блок или синдром ошибки. Пример реализации на языке описания аппаратуры Verilog представлен на рис. 3.

По авторскому алгоритму матричного деления на основании образующего полинома (выбран предыдущим модулем) данный модуль формирует таблицы вычисления остатков от деления, которые будут использоваться в процедурах кодирования и декодирования. На выходе модуля доступен сформированный быстродействующий блок вычисления контрольных разрядов и синдрома ошибок.

Модуль проектирования кодека

Для выбора структуры кодека системы проектирования проведено исследование [6], в результате которого выбрана и формализована структура быстродействующего кодека помехоустойчивых кодов (на рис. 5 представлена одна из возможных структур реализации). Для окончательной конфигурации кодека модуль генерирует таблицы декодирования.

На выходе модуля доступны все параметры кодека, необходимые для программной реализации. Для аппаратной реализации все вычисленные параметры передаются в модуль реализации на ПЛИС.

Модуль реализации кодека на ПЛИС

На заключительном этапе формируется проект, состоящий из файлов, содержащих программу, описывающую работу кодека на языке описания аппаратуры Verilog (на рис. 6 приведён пример

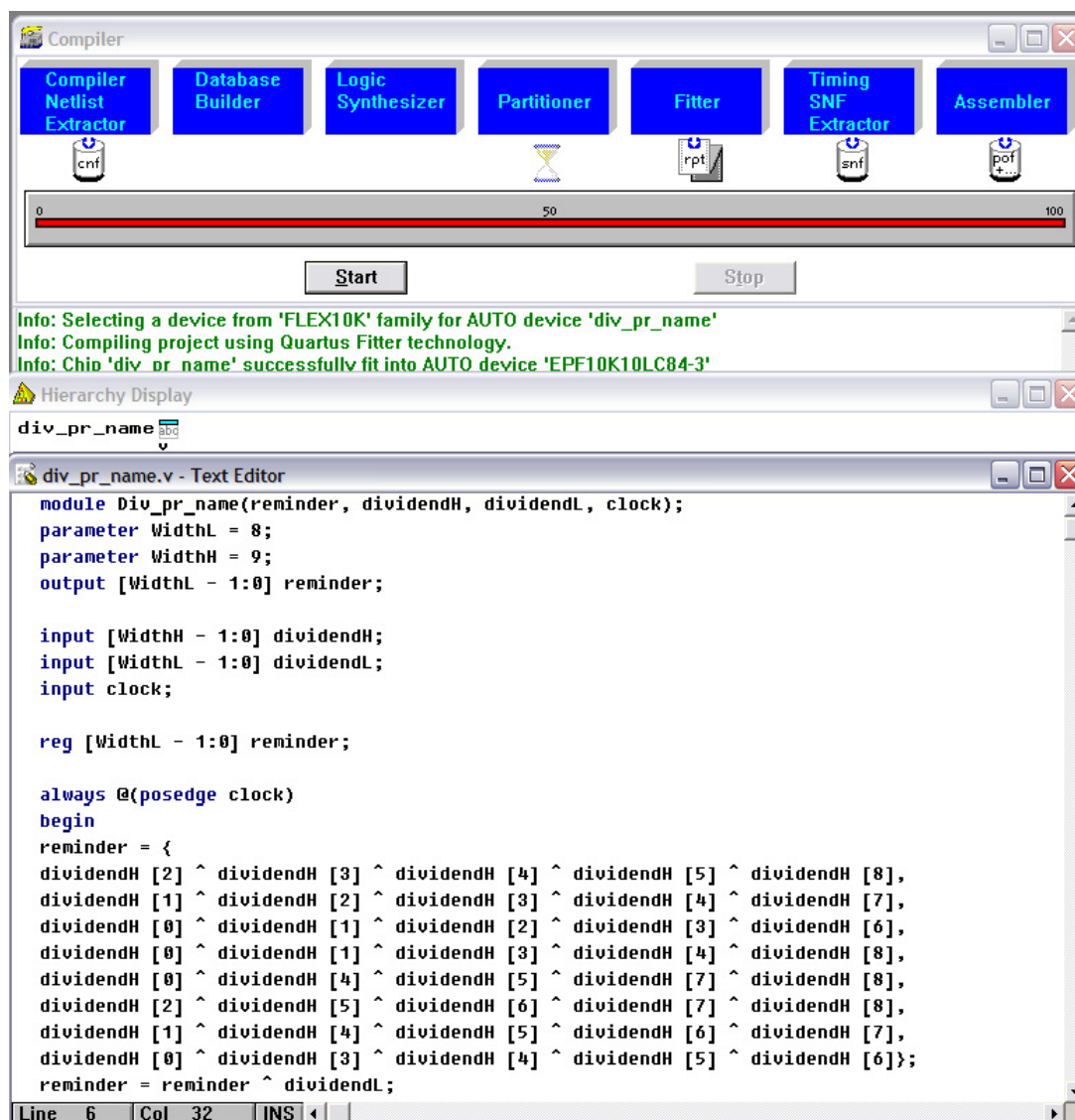


Рис. 3. Пример реализации системой матричного деления в САПР Altera Max+plusII

		Destination							
		remainder0	remainder1	remainder2	remainder3	remainder4	remainder5	remainder6	remainder7
S	clock	9.9ns	9.9ns	9.9ns	9.9ns	10.9ns	9.9ns	9.9ns	9.9ns

Рис. 4. Матрица задержек для схемы матричного деления

компиляции проекта). Формируется прошивка для ПЗУ, в которой будут содержаться таблицы декодирования. При генерации конфигурационного файла для ПЗУ применяется метод уменьшения требуемого объема памяти при табличном декодировании. Используется метод, подобный частично-ассоциативной организации кэш памяти. Затем готовый проект используется для конфигурирования ПЛИС (на рис. 9 приведены диаграммы задержек в режимах кодирования и декодирования), используя САПР фирмы Altera (Max+plus II, Quartus).

Из рис. 7 видно, что задержки в режимах кодирования и декодирования равняются 20 и 25 нс соответственно для кристалла EPF10K30ETC144-1 семейства ПЛИС Flex10KE фирмы Altera.

Заключение

Предлагаемая система позволит инженеру проектировать быстродействующие кодеки помехоустойчивых кодов [6] без углубленных знаний в области помехоустойчивого кодирования по двум

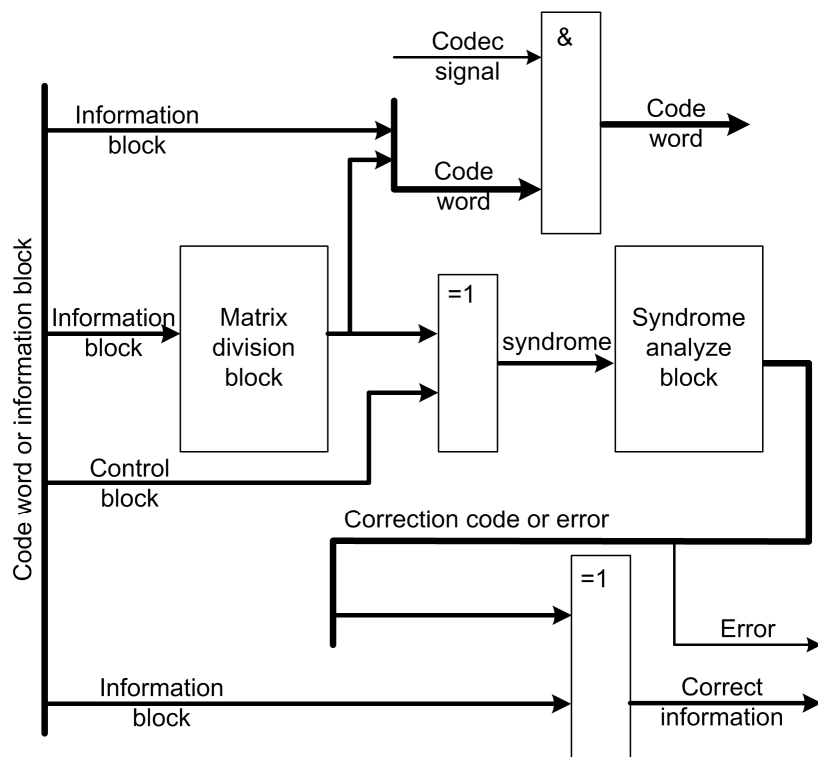


Рис. 5. Структура кодека

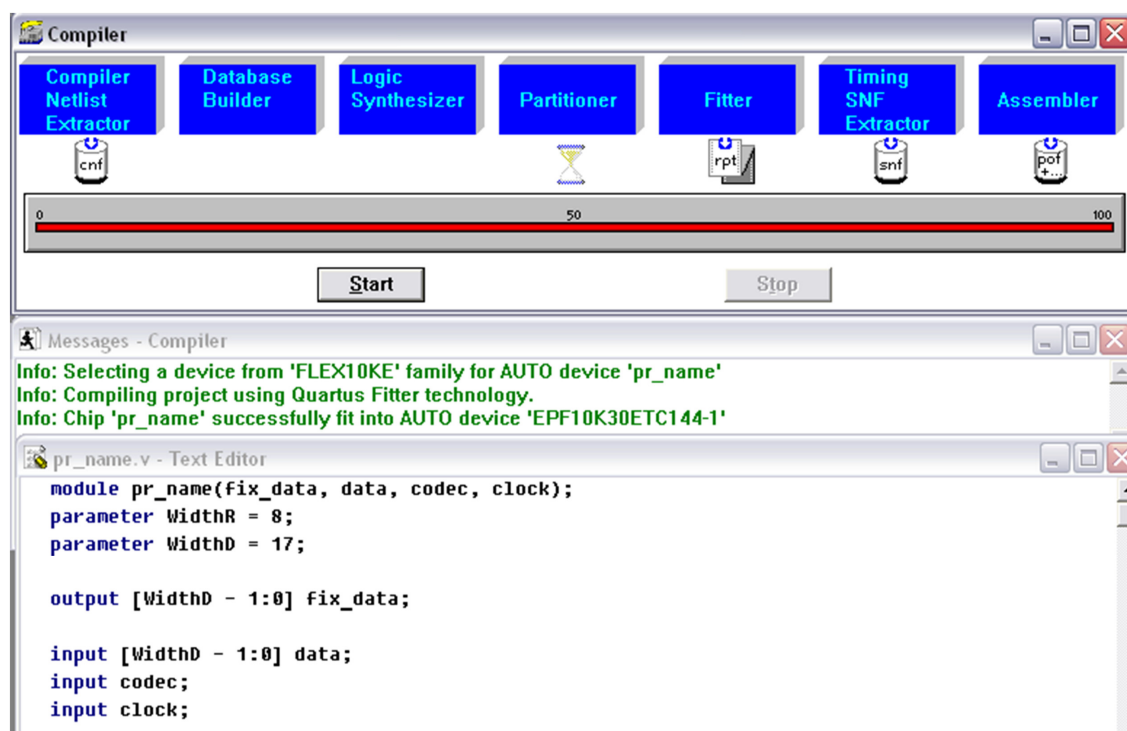


Рис. 6. Пример компиляции проекта в САПР Max+plusII

параметрам (длина информационного блока, корректирующая способность кода). На выходе системы доступны данные как для программной, так и для аппаратной реализации. В качестве аппаратной реализации формируются проектные файлы для конфигурирования ПЛИС фирмы Altera.

Помехоустойчивые кодеки, проектируемые системой, обладают следующими особенностями:

- быстродействие (кодирование 20 нс, декодирование 25 нс при длине кодового слова равной двум байтам);

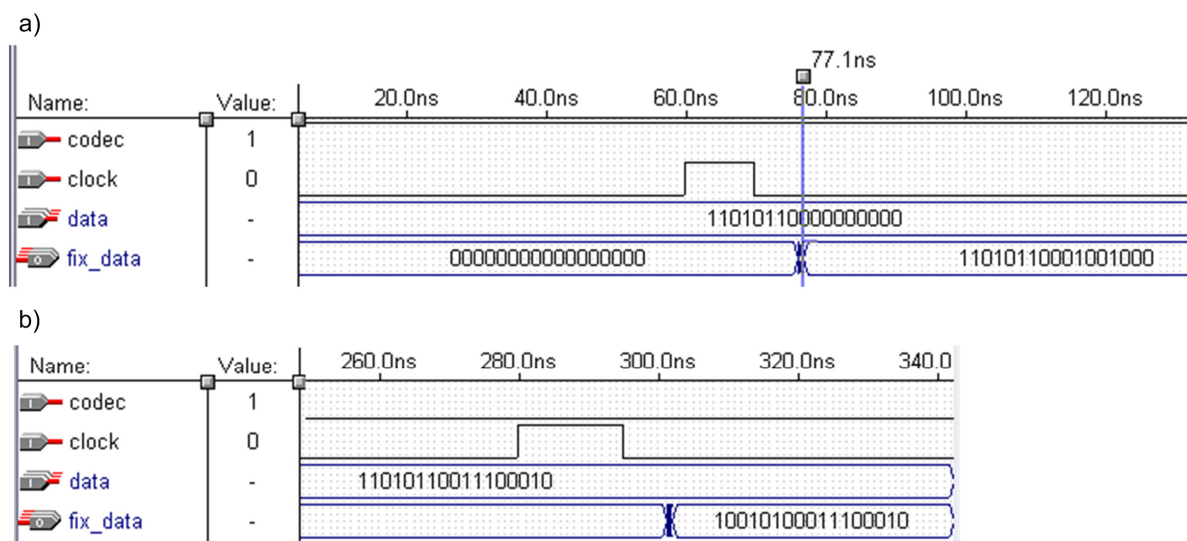


Рис. 7. Моделирование работы кодека в режимах: а) кодирования и б) декодирования

- длина кодового слова меньше, чем у БЧХ-кодов для случаев $t=2$, $m=\{1-4, 8-13, 22-24\}$, для $t=3$, $m=\{2, 6-13, 17-26\}$ и для $t=4$, $m=\{1-26\}$;
- возможность работать с данными при параллельном способе передачи (за счёт блока матричного деления);
- исправление пакетов ошибок в дополнение к исправлению независимых ошибок;

- сигнализация о нахождении неисправимой ошибки.

Авторы благодарят администрацию суперкомпьютерного кластера «СКИФ-политех» за содействие при проведении расчётов.

Работа выполнена на кафедре вычислительной техники института «Кибернетический центр» Томский политехнический университет и поддержана грантом «У.М.Н.И.К» Фонда содействия развитию малых форм предприятий в научно-технической сфере.

СПИСОК ЛИТЕРАТУРЫ

1. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. — М.: Техносфера, 2005. — 320 с.
2. Мак-Вильямс Ф.Дж., Слоэн Н.Дж.А. Теория кодов, исправляющих ошибки. Пер. с англ. И.И. Грушко, В.А. Зиновьева — М.: Связь, 1979. — 744 с.
3. Morelos-Zaragoza. BCH codes [Электронный ресурс]. — режим доступа: <http://www.eccpage.com/bch3.c>
4. Markus Grassl. On-line Tables of linear Codes [Электронный ресурс]. — режим доступа: <http://www.codetables.de>
5. Буркатовская Ю.Б., Мальчуков А.Н., Осокин А.Н. Быстродействующие алгоритмы деления полиномов в арифметике по модулю два // Известия Томского политехнического университета. — 2006. — Т. 309. — № 1. — С. 19–24.

6. Буркатовская Ю.Б., Мальчуков А.Н., Осокин А.Н. Быстродействующий кодек БЧХ-кодов на ПЛИС // Приборы и системы. Управление. Контроль. Диагностика. — 2006. — № 3. — С. 21–23.

Поступила 16.04.2008 г.

Ключевые слова:

Кодек, помехоустойчивый код, кодовое слово, система автоматизированного проектирования, ПЛИС.