# Министерство науки и высшего образованияРоссийской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Воронежский государственный технический университет»

Кафедра конструирования и производства радиоаппаратуры

# ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ УСТРОЙСТВ ОБРАБОТКИ СИГНАЛОВ

# МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ по дисциплине «Проектирование цифровых устройств обработки сигналов» для студентов направления 11.03.03 «Конструирование и технология электронных средств»(направленность «Проектирование и технология радиоэлектронных средств») всех форм обучения

ha(m)

Воронеж 2018

# Составители: канд. техн. наук Н. В. Ципина

Проектирование цифровых устройств обработки сигналов: методические указания к выполнению лабораторных работ по дисциплине «Проектирование цифровых устройств обработки сигналов» для студентов направления 11.03.03 «Конструирование и технология электронных средств» (профиль

«Проектирование и технология радиоэлектронных средств») всех форм обучения / ФГБОУ ВО «Воронежский государственный технический университет»; сост.: Н. В. Ципина. Воронеж: Изд-во ВГТУ, 2018. 17 с.

Основной целью указаний является выработка навыков работы с цифровыми устройствами обработки сигналов, уяснение их принципа действия, характеристик и параметров.

Предназначены для проведения лабораторных работ по дисциплине «Проектирование цифровых устройств обработки сигналов» для студентов 4 курса.

Методические указания подготовлены в электронном виде и содержатся в файле Лаб.раб\_ПроектЦУОС.pdf.

Табл. 3. Библиогр.: 16 назв.

УДК 621.3.049.7.002 (075) ББК 38.54

# Рецензент О. Ю. Макаров, д-р техн. наук, проф. кафедры конструирования и производства радиоаппаратуры ВГТУ

Издается по решению редакционно-издательского совета Воронежского государственного технического университета

### 1 ЦЕЛИ И ЗАДАЧ ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ

Лабораторные работы выполняются на 8 семестре обучения у бакалавров 11.03.03 Конструирование и технология электронных «Проектирование средств. Направленность и технология радиоэлектронных средств». При выполнении практических работ по дисциплине «Проектирование цифровых устройств обработки сигналов» студенты должны научиться правильно и творчески полученные использовать знания. ИМИ при прохождении «Электротехника дисциплин: электроника», теоретических И «Схемои системотехника электронных средств», «Материалы и Студенты компоненты электронных средств». должны ознакомиться с видами литературных источников и справочной информацией, необходимо которые использовать при проектировании печатных плат.

В результате выполнения практических работ они должны уметь решать следующие задачи:

осуществлять обзор литературных источников по заданной теме;

осуществлять поиск необходимой справочной информации по теме проекта;

выбирать необходимую элементную базу исходя из задания проектирования;

проводить необходимые при проектировании расчеты.

#### ОБЪЕМ ОТЧЕТА

Отчет по лабораторным работам должен содержать 5 – 8 страниц машинописного текста формата А4.

2. Содержание лабораторных занятий

На сегодняшний день представлено большое количество литературы, в ко торой изложены известные базовые алгоритмы цифровой обработки сигналов (ЦОС), однако реализация даже

простейших алгоритмов на базе микропроцессоров может вызвать трудности у программистов начального уровня квалификации. Это объясняется тем, что реализация алгоритма ЦОС тесно связана с устройством обработки сигналов, то есть для реализации алгоритма ЦОС необходимо знание архитектуры устройства обработки сигналов (микропроцессора, программируемой логической интегральной схемы и т. п.).

Таким образом, для начала освоения цифровых сигнальных процессоров необходимо выбрать устройство, имеющее достаточно высокое быстродействие, но в то же время обладающее не самой сложной архитектурой. Это даст возможность разработчику сосредоточиться на программной реализации алгоритмов ЦОС, не уделяя большого внимания вопросам сложного взаимодействия между ядром и периферийными устройствами микропроцессора.

Используемый тип микропроцессора, желательно, должен распространен рынке широко быть на И использоваться разработчиками оборудования в регионе. В настоящее время одним из наиболее известных производителей, выпускающих сигнальные процессоры, является компания Texas Instruments. Ее продукция получила широкое распространение в нашей стране. Кроме того, рядом российских компаний выпускаются семейства цифровых сигнальных процессоров, являющихся аналогами процессоров компании Texas Instruments. Цифровые сигнальные процессоры данного производителя представлены тремя семействами: С2000, С5000, С6000. Семейство сигнальных процессоров С2000 является наиболее ранним, пор поддерживается но ло сих И совершенствуется компанией. Это семейство включает процессоры, ориентированные на использование в системах промышленного контроля и управления. Процессоры сочетают в себе мощное вычислительное ядро и обширный набор периферийных устройств. Таким образом, использование процессоров данного семейства эффективно обработку позволяет выполнять информации, поступающей от внешних устройств различного назначения.

Кроме того, процесс разработки программного обеспечения для сигнальных процессоров семейства C2000 является достаточно простым для начинающих программистов. Компания-разработчик предоставляет наборы исходных кодов, которые могут быть использованы в качестве основы для создания программного обеспечения микропроцессоров на языке C++. Все сказанное выше свидетельствует о том, что семейство сигнальных процессоров C2000 является наиболее подходящим для начала освоения реализации алгоритмов ЦОС на базе цифровых сигнальных процессоров.

ПОДГОТОВКА РАБОЧЕГО МЕСТА К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

К выполнению лабораторных работ рекомендуется допускать студентов, прошедших предварительное обучение основам работы в программных средах Matlab и Code Composer Studio. Лица, выполняющие лабораторные работы, должны иметь навыки разработки программ на языке C++.

Необходимо соблюдать последовательность выполнения лабораторных работ в соответствии с их номерами.

Все лабораторные работы выполняются с использованием стенда ЦОС.Внешний вид стенда с расположением разъемов показан на рис. 1.



Перед выполнением лабораторных работ необходимо выполнить действия по подключению стенда ЦОС к ПЭВМ, а также настройку программной среды Code Composer Studio для начала

ХОД ПОДГОТОВКИ РАБОЧЕГО МЕСТА

разработки ПО.

1. Собрать рабочее место для выполнения лабораторной работы по одной из эквивалентных схем (рис. 2 или 3). Незадействованные приборы подключаются в процессе выполнения лабораторной работы.



*Puc.* 2



*Puc. 3* 

1. Подготовить к работе все приборы и ПЭВМ согласно их инструкциямпо эксплуатации.

2. Включить ПЭВМ.

3. Включить источник питания. При этом под прозрачной крышкой стендаЦОС должны загореться две лампы зеленого цвета.

Если для выполнения лабораторной работы используется эквивалент

1. Схема, показанная на рис. 3, следует пропустить п. 6–15 и перейти к п. 16.

2. Запустить программу «cc\_setup.exe» (ярлык «Setup CCStudio v3.3» на рабочем столе ПЭВМ). При этом на экране появится окно «Code Composer Studio Setup», представленное на рис. 4.

🐬 Code Composer Studio Set	up					= 🛛 🗙
File Edit View Help						
System Configuration	Augilable Eastern Reards	Family	Platform	Endianness	^	My System 🗠
- System semigaration	Available Pactory Boards	C28xx 🔻	ezdsp 🔻	Al 🔻		
📮 My System	F2804x eZdsp	C28xx	ezdsp	*		
	F2808 eZdsp	C28xx	ezdsp	*	-	
	F2812 eZdsp	C28xx	ezdsp	*		
	F28335 eZdsp	C28xx	ezdsp	*		
	R2812 eZdsp	C28xx	ezdsp	*		
					~	×
	🖼 Factory Boards 🖼 Custom B	oards   🏟 Crea	te Board			> >
Save & Quit Remove	< Add Add Multiple	]				Modify Properties
Drag a device driver to the left to a	dd a board to the system.					

Puc. 4

Если на панели System Configuration имеется

3. список выбранных ранее устройств, то он должен быть очищен. Для этого необходимо последовательно выбрать пункты меню «File» — «Remove All».

4. В окне «Code Composer Studio Setup» на панели Family в выпадающем меню выбрать пункт «C28xx», на панели Platform в выпадающем меню выбрать пункт «ezdsp». Вид окна после перечисленных действий представлен на рис. В окне «Code Composer Studio Setup» на панели Available Factory Boards двойным нажатием «мыши» выбрать пункт «F2812 eZdsp». При этом наименование выбранной отладочной платы должно появиться на панели System Configuration. Вид окна после перечисленных действий представлен на рис. б.



Рис. 6

5. В окне «Code Composer Studio Setup» на панели System Configuration нажатием правой кнопки «мыши» выбрать пункт «F2812 eZdsp» (рис. 7).

🐬 Code Compo	ser Studio Setu	ıp		= 🛚 🗶
File Edit View	Help			
System Configuration		Current Proccesor Type	Driver Location	cpu_0 ^
My System	REALER	-	C:\CCStudio_v3.3\drivers\sdgo2812eZdsp.dvr	Device Type: CPU
<b>A</b> 0	Rename	F2		
	Remove	Del		GEL File:
		i5		C.ICCSIdaid.
	Properties	Alt+Enter		
		🖼 Factory Boards 🔤	Custom Boards 🌸 Create Board	×
Save & Quit	Remove	<< Add << A	idd Multiple	Modify Properties
Drag a device drive	r to the left to ad	id a board to the system.		11

*Puc.* 7

В появившемся контекстном меню выбрать пункт «Properties...». При этом на экран будет выведено окно «Connection Properties». Вид окна после перечисленных действий представлен на рис. 8.

🌮 Code Composer Studio Setup		
File Edit Vi Connection Properties	🛛	
System Co Connection Name & Data File Connection Properties		cpu_0
My Systen     Connection     F2812 eZdsg     (SD510 Emulator)     Name:		Device Type: CPU
Auto-generate board data file		GEL File: C:\CCStudic
Browse		
Diagnostic Utility: Browse		~
Diagnostic Arguments:		>
Save & Du		Modify Properties
Drag a device driver to the left to add a board to the system.		11

*Puc.* 8

1. В окне «Connection Properties» нажать кнопку «Next». Вид окна послеперечисленных действий представлен на рис. 9.

🌍 Code Com	poser Studio Setup		
File Edit Vi	Connection Properties		
System Co	Connection Name & Data File Connection Properties	Γ	cpu_0 ^
My Systen	Property Value		Device Type: CPU
*	Emulator I/O Port PP Emulator address is 0x378 💌		GEL File: C:\CCStudic
	Change property value as necessary in the right column.	-	×
Save & Qui	Finish Отмена		Modify Properties
Drag a device d	river to the left to add a board to the system.	-	

Puc. 9

2. В окне «Connection Properties» проверить соответствие адреса в поле Value (рис. 9) адресу LPT-порта ПЭВМ, к которому подключен стенд ЦОС. Адреса LPT-портов ПЭВМ в операционных системах Windows 7/ХР можно узнать, открыв окно диспетчера устройств. При необходимости в выпадающем списке поля Value выбрать требуемый адрес LPT-порта ПЭВМ.

3. В окне «Connection Properties» нажать кнопку «Finish». Данное действие приведет к закрытию окна «Connection Properties» и возвращению в окно

«Code Composer Studio Setup».

4. В окне «Code Composer Studio Setup» нажать кнопку «Save & Quit» При этом на вопрос «Start Code Composer Studio on exit?» следует ответить

«Нет» («No»).

1. Запустить программу «сс\_арр.exe» (ярлык «CCStudio v3.3» на рабочем столе). Общий вид главного окна среды Code Composer Studio 3.3 представлен на рис. 10.



#### Puc. 10

2. Установить соединение со стендом ЦОС. Для этого в меню главного окна среды Code Composer Studio 3.3 последовательно выбрать пункты меню

«Debug»→«Connect». При этом на экран будет выведено окно «Disassembly» (рис. 11), что свидетельствует об успешном установлении соединения со стендом ЦОС.

1	F2812 eZdsp/cpu_0 - TM5320C28xx	- Code Composer Studio - Not Connected 📃 🖸 🛛
File	Edit View Project Debug GEL Op	tion Profile Tools DSP/BIOS Window Help
徻	📽 🖬   🕹 🖻 💼   🗠 🗠	💽   & % % % % % @ 8 (** ***) •*
	~	▼ 参 曲 茜 合 秀 ● 条
<b>5</b> 2	60 🗇 🗃 🛤 🔠 🔤 🗖	
74	J Eiles	Disassembly 😑 🛛 🗶
8	E GEL files	⇒BFFC00 28AD MOV @SP,1^
n.	Projects	3FFCU2 561F SETC OBJM(
		3FFC03 5616 CLRC AMODI 3FFC04 5616 SFTC MOMIN
0		3FFC05 2940 CLRC PAGE
0*		3FFC06 761F MOVW DP,#(
品		3FFC08 2902 CLRC OVM
5		3FFC09 FF69 SPM 0
		SFECOA 8D7F SUB ACC. Y
8-1-1-		

Puc. 11

Подготовку рабочего места к выполнению лабораторных работ

считать успешной при отсутствии сообщений об ошибках в ходе выполненных действий. Структурная схема стенда ЦОС приведена в приложении А. Схема электрическая принципиальная стенда ЦОС представлена в приложении Б.

Скопировать содержание лазерного диска ФЦКН.468369.024ЛД в корневой каталог на жестком диске ПЭВМ. Лазерный диск содержит исходные файлы для данного курса лабораторных работ

#### ЛАБОРАТОРНАЯ РАБОТА № 1

# ИЗУЧЕНИЕ АНАЛОГО-ЦИФРОВОГО ПРЕОБРАЗОВАТЕЛЯ МИКРОПРОЦЕССОРА TMS320F2812

Цель работы

Целью работы является изучение устройства и режимов работы аналогоцифрового преобразователя, входящего в состав микропроцессора TMS320F2812.

Задание к работе

1. Ознакомиться с устройством блока аналого-цифрового преобразователямикропроцессора TMS320F2812.

2. Изучить назначение регистров управления аналогоцифрового преобразователя микропроцессора TMS320F2812.

3. Разработать ПО, выполняющее аналого-цифровое преобразование сигналов с частотой дискретизации 10 кГц.

4. Произвести дискретизацию сигналов с помощью разработанного ПО.

5. Сделать выводы по результатам работы.

Пояснения к работе

Стенд ЦОС построен на базе отладочной платы TMDSEZS2812-0E, которая в свою очередь основана на микропроцессоре TMS320F2812.

Микропроцессор TMS320F2812 содержит блок аналогоцифрового преобразователя, позволяющий производить дискретизацию 16 независимых входных сигналов. Разрядность АЦП равна 12. Диапазон входных сигналов: 0–3 В [1, 2].

Аналоговый сигнал a(t), имеющий ограниченную ширину спектра, дискретизируется в отсчетные моменты времени, кратные шагу дискретизации  $\Delta t \Box$  T=const [3]. Результатом дискретизации по времени является получение последовательности значений (чисел)  $x(nT) \equiv x_n$ , где n = 0, 1, 2, ... Пример дискретного сигнала показан рис. 12.



#### Puc. 12

результате последующей процедуры квантования B формируется цифровой сигнал x<sub>п</sub>(*n*T). Отличительной особенностью замена значений дискретного квантования является сигнала дискретной шкалой h1, h2,..., hN. При этом разность между некоторому фиксированному уровнями кратна значению, называемому шагом квантования  $\Delta$  сигнала. Каждому уровню квантования дискретного сигнала ставится соответствие В определенный код, т. е. уровни квантования связываются с числами в выбранной системе счисления (рис. 13).

 $X_{II}(nT)$ 



Puc. 13

В технике, как правило, используются целые числа в двоичной системе счисления с ограниченным количеством разрядов [3]. При этом значения кодов  $x_u(nT)$  рассматриваются как значения дискретного сигнала x(nT).

Согласно документу [1] соотношение между напряжением на входе и значением цифрового кода, формируемого АЦП микропроцессора TMS320F2812, определяется выражением

На рис. 14. показано содержание регистра ADCTRL1.

15	14	13	12	11	10	9	8	
Reserved	RESET	SUSMOD1	SUSMOD0	ACQ PS3	ACQ PS2	ACQ PS1	ACQ PS0	
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
7	6	5	4	3			0	
CPS	CONT RUN	SEQ1 OVRD	SEQ CASC	Reserved				
R/W-0	R/W-0	R/W-0	R/W-0	R-0				

Puc. 14

На данном рисунке: R – данный бит предназначен только для чтения; R/W – данный бит предназначен для чтения и записи; -0 – нулевое значение после сброса.

Бит 15 зарезервирован.

Бит 14 (RESET) – программный сброс модуля АЦП. Запись

нулевого значения в этот бит не дает эффекта, при записи единичного значения производится программный сброс модуля АЦП.

Биты 12–13 (SUSMOD0–SUSMOD1) определяют реакцию модуля АЦП на различные события эмулятора (например, достижение точки останова программыи т. п.).

Биты 8–11 (ACQ\_PS0–ACQ\_PS3) определяют время ожидания окончания процесса аналого-цифрового преобразования.

Бит 7 (CPS) устанавливает масштабирующий коэффициент частоты тактирующего сигнала АЦП [1].

Бит 6 (CONT RUN) определяет режим запуска аналогоцифрового преобразования:

1 – режим «Continuous conversion» (после завершения текущего аналогоцифрового преобразования следующее преобразование может быть запущено без выполнения сброса модуля АЦП) [1];

0 – режим «Start-stop» [1].

Бит 5 (SEQ OVRD) устанавливает режим последовательности выполнения аналого-цифрового преобразования [1]

Бит 4 (SEQ CASC) устанавливает режимы выполнения аналого-цифровогопреобразования [1]:

Биты 0-3 зарезервированы.

Регистр ADC Control Register 2 (ADCTRL2) в основном предназначен для управления прерываниями, формируемыми модулем АЦП [1].

На рис. 15 показано содержание регистра ADCTRL3.



На данном рисунке: R – данный бит предназначен только для чтения; R/W – данный бит предназначен для чтения и записи; -0 – нулевое значение после сброса.

Биты 9–15 зарезервированы.

Бит 8 (EXTREF) разрешает или запрещает использование внешних опорных источников напряжения.

Биты 6–7 (ADCBGRFDN[1:0]) предназначены для управления питанием цепи опорного напряжения.

Бит 5 (ADCPWDN) предназначен для управления питанием аналоговых цепей модуля АЦП.

Биты 1–4 (ADCCLKPS[3:0]) устанавливают масштабирующий коэффициент частоты тактирующего сигнала АЦП [1, 2].

Бит 0 (SMODE SEL) устанавливает синхронный или последовательный режим аналого-цифрового преобразования [1].

На рис.16 показано содержание регистра ADCRESULTO. В данном регистре хранится текущий результат аналого-цифрового преобразования канала АЦП ADCINA0. Содержание регистров текущего результата аналого-цифрового преобразования всех каналов АЦП одинаково [1].

15	14	13	12	11	10	9	8
D11	D10	D9	D8	D7	D6	D5	D4
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
7	6	5	4	3	2	1	0
D3	D2	D1	D0	Reserved	Reserved	Reserved	Reserved
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

Puc.	1	6
------	---	---

На данном рисунке: R – данный бит предназначен только для чтения; -0 – нулевое значение после сброса.

Биты 4–15 содержат 12-разрядный цифровой код результата аналого-цифрового преобразования (бит 4 – младший разряд кода, бит 15 – старший разряд кода).

Биты 0-3 зарезервированы.

АЦП микропроцессора TMS320F2812 имеет еще ряд регистров, предназначенных для установки режимов аналогоцифрового преобразования, управления последовательностью опроса каналов АЦП и т. д. Это регистры ADCMAXCONV, ADCASEQSR, ADCST, ADCCHSELSEQ1 и др. [1].

Пример:

Установить следующие параметры модуля АЦП: режим «Continuous conversion», режим «Cascaded mode», частота тактирующего сигнала модуля АЦП равна HSPCLK/2, где HSPCLK – частота тактирующего сигнала для высокоскоростных устройств [4].

Для установки указанных параметров в исходном коде ПО для микропроцессора TMS320F2812 должны быть выполнены

следующие команды [5]:// ADC module clock = HSPCLK/2\*ADC\_CKPS= HSPCLK/2\*1.

AdcRegs.ADCTRL3.bit.ADCCLKPS = 1;

//Режим Cascaded mode. AdcRegs.ADCTRL1.bit.SEQ\_CASC = 1; // Режим Continuous conversion. AdcRegs.ADCTRL1.bit.CONT\_RUN = 1;

Чтение результата аналого-цифрового преобразования для канала ADCINA0 может быть выполнено следующим образом:

Uint16 U; U=(AdcRegs.ADCRESULT0>>4);

В большинстве практических случаев требуется, чтобы аналого-цифровое преобразование выполнялось не один раз, а многократно через промежуток времени, равный шагу дискретизации сигналов Т. В связи с этим возникает необходимость использования одного или нескольких таймеров, имеющихся в составе микропроцессора TMS320F2812 [4, 6].

Одним из вариантов организации периодического выполнения преобразования аналогоцифрового является использование формируемых таймерами. Таймер прерываний. должен быть настроен на отсчет интервала времени, равного шагу дискретизации сигнала. После истечения данного промежутка времени таймер формирует прерывание. По прерыванию микропроцессор перезапускает таймер и дает команду блоку АЦП на выполнение аналого-цифрового преобразования. После этого микропроцессор режиме ожидания до тех пор, пока таймер не находится В сформирует новое прерывание.

Более подробную информацию о таймерах микропроцессора TMS320F2812 можно найти в работе [6].

Ход работы

1. Выполнить действия п. 1–18 главы «Подготовка рабочего места к выполнению лабораторных работ».

2. Открыть проект ПО лабораторной работы №1. Для этого в меню главного окна среды Code Composer Studio 3.3 последовательно выбрать пункты меню «Project»→ «Open…».

3. B (рис. 17) выбрать появившемся окне файл «Example\_ADC.pjt» из папки «X:\ADC\DSP281x\_examples\ADC» и нажать кнопку «Открыть». Здесь Х – условное обозначение литеры содержащего исходные файлы для данного диска, курса лабораторных работ. При этом в левой части главного окна средыCode Composer Studio 3.3 появится «дерево файлов» проекта, отображающее его структуру (рис. 18) [7].

Project Open			2 🛛
Папка: ն	ADC	• ÷ •	📸 💷 -
Debug Example_4	vDC.CS_ vDC.p)t		
Имя файла:	Example_ADC		Открыть
Тип файлов:	Project Files (*.pit)	•	Отмена
			Справка

Puc. 17

1 😻	'F2812 eZd≤	sp/cpu_(	0 - TM53	20028x	к - Со	ide Comp	oser	Studio						
File	Edit View	Project	Debug	GEL C	ption	Profile	Tools	DSP/BIC	G Windo	w He	þ			
徻	iii 🗐 🔛	X Peri	<b>B</b>   10	$\simeq  $	_				•	8 A	- Tio Ro 🖓	4日	ë : <b>№</b>	- 🏀 🛛 🐠
			Y			Ψ	0	曲曲	*	0 2	چ ک			
چي	60 🚨		8 🗉 🖂		<u> </u>	8								
74	0	_	-	-		Disassen	ibly							
10		EL files		N		♦BFFC	00	28AD	MOV		@SP, #^			
8*	P	rojects		- N		3FFC	02	561F	SETC		OBJM(			
- {}+	i de da	🖥 Examı	ple_ADC	.pjt 🔰		3FFC	03	5616	CLRC		AMODI			
6				1		3FFC	04	561A	SETC		MOM11			
መ						3FFC	05	2940	CLRC		PAGE			
Ľ.						3FFC	06	761F	MOVW		DP,#(			
韵						3FFC	08	2902	CLRC		OVM			
						3FFC	09	FF 69	SPM		0			
0					1	SEEC	ΠA	8D7F	SHB		ACC. 3			
	l 비스	J					_			_	2 //			
8	HALTED									F	or Help, press	F1		

Puc. 18

4. Перемещаясь по «дереву файлов» выбрать двойным щелчком «мыши» файл «Example\_281xSpi\_FFDLB.c» (рис. 19). Вид окна после перечисленных действий представлен на рис. 20.

Полный текст подпрограммы

«Example\_281xSpi\_FFDLB.с» для лабораторной работы №1 представлен в приложении В.



Puc. 19



Puc. 20

5. Изучить текст подпрограммы, находящийся в файле «Example\_ 281xSpi\_FFDLB.c».

6. Выполнить компиляцию проекта. Для этого в главном окне среды Code Composer Studio 3.3 нажать кнопку В группе Собщений об ошибках в процессе компилированным при отсутствии сообщений об ошибках в процессе компиляции.

7. Запрограммировать микропроцессор. Для этого в меню главного окна среды Code Composer Studio 3.3 последовательно выбрать пункты «Tools»→

«F28xx On-Chip Flash Programmer». При этом на экран

# будет выведено окно «On-Chip Flash Programmer» (рис. 21).

On-Chip Flash Programmer	×
Clock Configuration OSCCLK (Mhz): 30.000 CLKINDV (/2) PLLCR Value: 10 SYSCLKOUT (MHz); 150.0000	Einse Sector Selection     ✓ Sector S (Sector SFFFF)     ✓ Sector F: (SE4000.3E7FFF)     ✓ Sector F: (SE4000.3E7FFF)     ✓ Sector C: (S7000.3E7FFF)
Code Security Pareneed           Key 7 (Duble?)         FFFF           Key 6 (DubleS)         FFFF           Key 5 (DubleS)         FFFF           Key 3 (Duble2)         FFFF           Key 3 (Duble2)         FFFF           Key 1 (Duble2)         FFFF	Openation Please specify the COFF file to Program.Verify: C.L   Erase, Program, Verify Pro
Program Password Flash Programmer Settings	C Load RAM Only Flash+OTP: Execute Operation Help

Puc. 21

8. В окне «On-Chip Flash Programmer» нажать кнопку «Browse...».

9. B появившемся окне (рис. 22) выбрать файл «Example ADC.out» папки ИЗ «X:\ADC\DSP281x\_examples\ADC\Debug» И нажать кнопку «Открыть». Здесь X – условное обозначение литеры диска, содержащего исходные файлы для данного курса лабораторных работ.

Открыть			<u>? ×</u>
Папка: 🗀	Debug	• <del>•</del> •	r 📰 🕶
Example_4	ADC.out		
Имя файла:	Example_ADC.out		Открыть
Тип файлов:	COFF Files (*.out)	V	Отмена

Puc. 22

В окне «On-Chip Flash Programmer» нажать кнопку «Flash Programmer Settings...», что приведет к появлению окна «Flash Programmer Settings» (рис. 23).

1	
T	٠

Select DSP Device to Program           © F2810         © F2801           © F2811         © F2802           © F2812         © F2806           © F2808         © F2808	Options C Load Symbols Display Tooltips Display Diagnostics	OK Cancel Help
Select version of Flash API Interface fi	ile:	
C:\CCStudio_v3.3\plugins\Flash28xx	\Algorithms\2812\FlashAPIInterface2812V2	Browse

Puc. 23

10. В окне «Flash Programmer Settings» нажать кнопку «Browse...».

11. В появившемся окне (рис. 24) выбрать файл

«FlashAPIInterface2812V2\_ 10.out» и нажать кнопку «Открыть».

Открыть			<u>? ×</u>
Папка:		▼ ← €	💣 🎟 •
🗐 FlashAPIIr	terface2812V2_10.out		
Имя файла:	FlashAPIInterface2812V2_10.out		Открыть
Тип файлов:	COFF Files (".out)	•	Отмена

Puc. 24

12. В окне «Flash Programmer Settings» нажать кнопку «ОК».

13. В окне «On-Chip Flash Programmer» нажать кнопку «Execute Operation», что приведет к запуску операций программирования микропроцессора.

ВНИМАНИЕ! ЗАПРЕЩАЕТСЯ ПОДАВАТЬ КАКИЕ-ЛИБО СИГНАЛЫ НА ВХОД СТЕНДА ЦОС ВО ВРЕМЯ ПРОГРАММИРОВАНИЯМИКРОПРОЦЕССОРА

ВНИМАНИЕ! ЗАПРЕЩАЕТСЯ ОТКЛЮЧАТЬ ПИТАНИЕ СТЕНДА ЦОС ВО ВРЕМЯ ПРОГРАММИРОВАНИЯ МИКРОПРОЦЕССОРА

Микропроцессор считать успешно запрограммированным при отсутствии сообщений об ошибках в процессе программирования. После успешного завершения процедуры программирования следует закрыть окно «On-Chip Flash Programmer».

14. Запустить программную среду Matlab, в командной строке среды Matlab выполнить команду:

>> InputSamples=0;

15. В программной среде Matlab раскрыть содержание переменной

«InputSamples» в редакторе массивов (рис. 30) [8].

File	Edit	View	Web	Window	Help
×	Ēð		Nume	ric forma	t shortG 💌 Size: 1 by 1
			1		
		1		0	

Puc. 30

16. Вставить скопированный в п. 23 столбец в переменную

«InputSamples». Для этого в окне редактора массивов программной среды Matlab нажать **правую** кнопку «мыши». В появившемся контекстном менювыбрать пункт «Paste» (рис. 31).

File E	Edit	View	/ Web	Window	Help					
X	Îħ	<b>n</b> .	Num	eric forma	t: shortG	•	Size: 1	by 1		
			1					 	 	
		1		Cut						
				Сору						
				Paste						

Puc. 31

17. Отобразить на графике дискретную последовательность отсчетов сигнала, поступивших с АЦП микропроцессора TMS320F2812. Для этого в командной строке среды Matlab выполнить команды:

>> plot(InputSamples, '-o');

>> ylabel('КОД')

>> xlabel('n')

Пример результата выполнения команд показан на рис. 32.



Puc. 32

18. Проанализировав содержание построенного графика, убедиться, что последовательность дискретных значений имеет гармоническую форму и на один период последовательности приходится не менее 10 дискретных значений. Значения последовательности должны быть целыми числами и лежать в

диапазоне от 0 до 4095.

Контрольные вопросы

1. Приведите последовательность команд на языке C++, необходимую для установки параметров и запуска аналогоцифрового преобразователя микропроцессора TMS320F2812. Поясните назначение каждой приведенной команды.

2. Приведите формулировку теоремы Котельникова, изложите ее сущность.

3. Из каких условий выбирается частота дискретизации аналоговых сигна-

лов?

4. Приведите последовательность команд на языке С++, необходимую для

установки параметров и запуска таймера микропроцессора TMS320F2812. Поясните назначение каждой приведенной команды.

5. Перечислите основные форматы представления чисел в вычислительной технике.

Как производится кодирование отрицательных чисел?

6. Какой формат представления чисел используется в микропроцессоре TMS320F2812?

7. Какую разрядность имеет встроенный аналого-цифровой преобразователь микропроцессора TMS320F2812?

8. Назовите основные типы аналого-цифровых преобразователей, выпускаемых промышленностью в настоящее время.

9. Назовите источники погрешности аналого-цифровых преобразовате-

лей.

#### ЛАБОРАТОРНАЯ РАБОТА № 2

# ИЗУЧЕНИЕ ВЗАИМОДЕЙСТВИЯ МИКРОПРОЦЕССОРА TMS320F2812C ЦИФРОАНАЛОГОВЫМ ПРЕОБРАЗОВАТЕЛЕМ

Цель работы

Целью работы является разработка ПО для микропроцессора TMS320F2812, выполняющего взаимодействие с цифроаналоговым преобразователем.

Задание к работе

1. Ознакомиться с устройством блока интерфейса SPI микропроцессораTMS320F2812.

2. Изучить назначение регистров управления блока интерфейса SPI микропроцессора TMS320F2812.

3. Разработать ПО, выполняющее передачу данных по интерфейсу SPI от микропроцессора TMS320F2812 к цифроаналоговому преобразователю.

4. Выполнить формирование сигналов с помощью цифроаналогового преобразователя.

5. Сделать выводы по результатам работы.

Пояснения к работе

В состав микропроцессора TMS320F2812 входит модуль последовательного интерфейса SPI (Serial Peripheral Interface) [4, 2]. Он предназначен для подключения различного периферийного оборудования (АЦП, ЦАП, микросхем flash-памяти и т. д.). Интерфейс SPI позволяет осуществлять высокоскоростную синхронную передачу данных между микропроцессором и некоторым периферийным устройством либо между несколькими микропроцессорами. В некоторых случаях этот интерфейс (как и интерфейс SCI) может быть использован для внутрисхемного программирования микропроцессора [9].

Стенд ЦОС оснащен 12-разрядным ЦАП, который принимает кодовые посылки от микропроцессора TMS320F2812 по интерфейсу SPI.

Интерфейс SPI является синхронным и имеет раздельные линии входных и выходных данных. Частота синхронизации может достигать 5 МГц [9]. В основном варианте использования предполагается, что интерфейс соединяет одно ведущее устройство с одним или несколькими ведомыми устройствами [9]. Наличие нескольких ведущих устройств на шине не допускается. Данный интерфейс может работать в режимах односторонней и двусторонней передачи данных.

Один такт синхронизации позволяет одновременно передавать и приниматьодин бит данных.

В интерфейсе используются три обязательных сигнала [9]:

– SCK (Serial Clock) – синхросигнал, которым ведущее устройство стробирует каждый бит данных;

– MOSI (Master Output Slave Input) – выходные данные ведущего устройства и входные данные ведомого устройства;

– MISO (Master Input Slave Output) – входные данные ведущего устройстваи выходные данные ведомого устройства.

Кроме того, для выбора ведомого устройства, которому передаются данные, может использоваться сигнал выбора устройства CS (Chip Select). Данные передаются в линию старшим битом вперед. Устройства различных производителей могут использовать разные режимы обмена, которые различаются фазой и полярностью синхросигналов.

Для настройки режима обмена модуля интерфейса SPI микропроцессора TMS320F2812 необходимо сконфигурировать три

регистра: SPICCR, SPICTL, SPIBRR [10].

На рис. 33 показано содержание регистра SPICCR (SPI Configuration Control Register). Данный регистр конфигурирует блок интерфейса микропроцессора (устанавливает полярность тактирующего сигнала, количество бит одного символа и т. д.).



Puc. 33

На данном рисунке: R – данный бит предназначен только для чтения; R/W – данный бит предназначен для чтения и записи; -0 – нулевое значение после сброса.

Бит 7 (SPI SW Reset) – программный сброс (при изменении конфигурации данный бит должен быть сброшен путем записи в него единичного значения):

0- данное значение бита должно быть установлено перед изменением параметров модуля интерфейса SPI;

– данное значение бита должно быть установлено после завершения изменения параметров модуля интерфейса SPI, т. е. перед выполнением передачи(приема) данных.

Бит 6 (CLOCK POLARITY) определяет полярность тактирующего сигнала:

1 – выдача данных производится по спаду сигнала SPICLK, прием данных – пофронту сигнала SPICLK;

0 – выдача данных производится по фронту сигнала SPICLK, прием данных –по спаду сигнала SPICLK.

Бит 5 зарезервирован.

Бит 4 (SPILBK) устанавливает программную посылку выходных сигналов интерфейса на его вход. Данная возможность может использоваться в режиме отладки ПО для тестирования интерфейса SPI.

Биты 0–3 (SPICHAR0–SPICHAR3) устанавливают разрядность передаваемого символа. В табл. 1 приведены значения рассмотренных битов, соответствующие различной разрядности символа (Character Length).

SPI CHAR3	SPI CHAR2	SPI CHAR1	SPI CHAR0	Character Length
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10
1	0	1	0	11
1	0	1	1	12
1	1	0	0	13
1	1	0	1	14
1	1	1	0	15
1	1	1	1	16

Таблица 1

На рис. 34 показано содержание регистра SPICTL (SPI Operation Control Register). Данный регистр устанавливает режим ведущего/ведомого, разрешает или запрещает передачу данных микропроцессором, выполняет управление формированием прерываний [10, 5].

7	6	5	4	3	2	1	0
	Reserved		OVERRUN INT ENA	CLOCK PHASE	MASTER/ SLAVE	TALK	SPI INT ENA
	R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Puc. 34

На данном рисунке: R – данный бит предназначен только для чтения; R/W – данный бит предназначен для чтения и записи; -0 – нулевое значение после сброса.

Биты 5-7 зарезервированы.

Бит 4 (Overrun INT ENA) разрешает или запрещает формирование прерыванийпо значению бита RECEIVER OVERRUN Flag bit регистра SPIST:

1 – прерывания по значению бита RECEIVER OVERRUN Flag bit разрешены; 0 – прерывания по значению бита RECEIVER OVERRUN Flag bit запрещены.

Бит 3 (CLOCK PHASE) устанавливает начальную фазу сигнала SPICLK:1 – тактирование с задержкой [10];

0-нормальный режим тактирования [10].

Бит 2 (MASTER / SLAVE) определяет режим работы модуля:1 – режим ведущего (Master);

0-режим ведомого (Slave).

Бит 1 (TALK) разрешает или запрещает передачу данных модулем SPI микропроцессора:

1– передача данных модулем SPI процессора разрешена; 0 – передача данных модулем SPI процессора запрещена.

Бит 0 (SPI INT ENA) разрешает или запрещает формирование

прерываний модулем SPI микропроцессора (кроме прерываний по значению бита RECEIVEROVERRUN Flag bit регистра SPISTS):

На рис. 35 показано содержание регистра SPIBRR (SPI Baud Rate Register). Данный регистр содержит кодовое значение скорости передачи информации модуля SPI процессора.



#### Puc. 35

На данном рисунке: R – данный бит предназначен только для чтения; R/W – данный бит предназначен для чтения и записи; -0 – нулевое значение после сброса.

Бит 7 зарезервирован.

Биты 0-6 (SPI BIT RATE 0- SPI BIT RATE 6) определяют значение скоростипередачи информации по интерфейсу SPI [10].

Интерфейс SPI имеет еще ряд регистров, предназначенных для управления режимами приема/передачи. Это регистры SPIFFTX, SPIFFRX, SPIFFCT, SPIPRI, SPISTS и др. Передача и прием посылок также осуществляются с использованием регистров. Более подробную информацию о содержании регистров интерфейса SPI можно найти в работе [10].

Пример:

Установить следующие параметры модуля интерфейса SPI: разрядность передаваемого символа: 12, выдача данных производится по спаду сигнала SPICLK, режим ведущего, нормальный режим тактирования, прерывания запрещены.

Для установки указанных параметров в исходном коде ПО для микропроцессора TMS320F2812 должны быть выполнены

следующие команды [5]:

// Бит 7 регистра SPICCR имеет значение 0,

// выдача данных по спаду сигнала SPICLK,

// разрядность символа 12 бит.SpiaRegs.SPICCR.all =0x004B;

// Режим ведущего,

// нормальный режим тактирования,

// передача данных модулем SPI процессора разрешена,

// все прерывания запрещены

SpiaRegs.SPICTL.all =0x0006;

// SPI Baud Rate = LSPCLK/(4+1)SpiaRegs.SPIBRR =0x4;

// Программный сброс после установки всех параметров SpiaRegs.SPICCR.all =0x00CB;

// Свободный запуск SPI после

// обработки событий эмулятораSpiaRegs.SPIPRI.bit.FREE = 1;

Для отправки посылки на передачу используется регистр SPITXBUF, дляприема – регистр SPIRXBUF.

Передача данных (например, числа 2254) может быть выполнена следующей командой:

SpiaRegs.SPITXBUF=2254;

Данная команда инициирует выдачу синхросигнала микропроцессором. На рис. 36 показан пример осциллограммы передачи 12-разрядного целого числа 2254 (100011001110 в двоичной системе) микропроцессором TMS320F2812. При этом используется выдача данных по спаду и нормальный режим тактирования (бит CLOCK POLARITY = 1, бит CLOCK PHASE = 0).



Puc. 36

На данном рисунке SPICLK – синхросигнал, SPISIMOA – выходные данныемикропроцессора TMS320F2812.

Так как интерфейс SPI является синхронным, прием данных ведомого устройства осуществляться может ОТ только при одновременной передаче данных от ведущего устройства (выдача происходит при синхросигнала только выполнении команды передачи данных ведущим устройством). Например, чтобы принять 12-разрядный символ от ведомого устройства, ведущее устройство должно отправить 12-разрядный символ. Вопросы приема данных по интерфейсу SPI ведущим устройством не относятся к предмету изучения данной лабораторной работы.

Ход работы

1. Выполнить действия п. 1–18 главы «Подготовка рабочего места к выполнению лабораторных работ».

2. Открыть проект ПО лабораторной работы № 2. Для этого в меню главного окна среды Code Composer Studio 3.3 последовательно выбрать пункты меню «Project»→ «Open…».

3. B (рис. 37) выбрать файл появившемся окне «Example DAC.pjt» из папки «X:\DAC\DSP281x examples\DAC» и нажать кнопку «Открыть». Здесь Х – условное обозначение литеры содержащего исходные файлы для данного курса лиска. лабораторных работ. При этом в левой части главного окна средыCode Composer Studio 3.3 появится «дерево файлов» проекта, отображающее его структуру (рис. 38) [7].

Project Open			? 🗙
Папка: 🗀	DAC	E (	* 💷 *
Debug Example_D Example_D	AC.CS_ AC.pjt		
Имя файла:	Example_DAC		Открыть
Тип файлов:	Project Files (*.pjt)	•	Отмена
			Справка

*Puc. 37* 

1 🐌	F281	2 eZds	;p/cpu_(	) - TM53	20028	жя - С	ode Con	nposer	Studio				- 🛛 🗙
File	Edit	View	Project	Debug	GEL	Option	n Profile	Tools	DSP/BIC	S Window	Help		
徻	<b>2</b>		x Po I	B   P	$\simeq$					▼ 6	" Na 15a 5a 🕯	• %   æ ē ∈ №	• 🐟 🕤
				-			1	- 8	· 🕮 🖽	▲  ⊙	2 🔍 🖉		
5	60	đ	1 🖪 🗷			<u>et</u>	8					_	
2	1776	~			_		Disasse	embly			= 🛛 🗙		
	ž	nies	FL filer				⇒ 3FF	C00	28AD	MOV	@SP, #^		
8*	6	- 🔄 Pr	rojects			1	3FF	C02	561F	SETC	OBJM		
P.	K	÷.,,	🔓 Examp	le_DAC	pjt	/	ЗFF	C03	5616	CLRC	AMODI		
6					1		ЗFF	C04	561A	SETC	MOM1P		
ក		~					ЗFF	C05	2940	CLRC	PAGE		
							3FF	C06	761F	MOVW	DP,#(		
閥							3FF	C08	2902	CLRC	OVM		
8							3 F F	C09	FF 69	SPM	0		
ē	<u>_</u> [	1⁄4	J			<	995	CUA	8575	SUB	Arr. 7		
<b>1</b>	O HA	LTED									For Help, pres	s F1	

Puc. 38

Перемещаясь по «дереву файлов», выбрать двойным щелчком «мыши» файл «Example\_281xSpi\_FFDLB.c» (рис. 39). Вид окна после перечисленных действий представлен на рис. 40. Полный текст подпрограммы

«Example\_281xSpi\_FFDLB.с» для лабораторной работы №2 представлен в приложении Г.



Puc. 39

4. Изучить текст программы, находящийся в файле «Example\_281xSpi\_ FFDLB.c».



Puc. 40

5. Выполнить компиляцию проекта. Для этого в главном окне среды Code Composer Studio 3.3 нажать кнопку 🛗 в группе 🕼 🛗 🛗 . Проект считатьуспешно

откомпилированным при отсутствии сообщений об ошибках в процессе компиляции.

6. Запрограммировать микропроцессор. Для этого в меню главного окна среды Code Composer Studio 3.3 последовательно выбрать пункты «Tools»→

«F28xx On-Chip Flash Programmer». При этом на экран будет выведено окно

«On-Chip Flash Programmer» (рис. 41).

On-Chip Flash Programmer	×
Clock Configuration	Erase Sector Selection
OSCCLK (Mhz): 30.000	Sector A: (3F6000-3F7FFF) Sector F: (3E4000-3E7FFF)
	Sector B: (3F4000-3F5FFF) Sector G: (3E0000-3E3FFF)
	Sector C: (3F0000-3F3FFF) Sector H: (3DC000-3DFFFF)
PLLUH Value: 10 •	Sector D: (3EC000-3EFFFF) Sector I: (3DA000-3DBFFF)
SYSCLKOUT (MHz): 150.0000	Sector E: (3E8000-3EBFFF) Sector J: (3D8000-3D9FFF)
Code Security Password	Operation
Key 7 (0xAE7): FFFF	Please specify the COFF file to Program/Verify:
Key 6 (0xAE6): FFFF	C:V Browse
Key 5 (0xAE5): FFFF	Erase, Program, Verify C Depletion Recovery
Key 4 (0xAE4): FFFF	C Erase Only C Frequency Test
Key 3 (0xAE3): FFFF	O Program, Verify Register: GPAMux
Keu 2 (0v6E2)	Program Only Pin: PWM1 (0)
	C Verify Only
Key1 (UxAE1): FFFF	Flash Random Wait State: 15 💌 O Calculate Checksums
Key 0 (0xAE0): FFFF	Flash Page Wait State: 15 💌 Elash:
Unlock Lock	OTP Wait State: 31 V OTP
Program Password	C Load RAM Only Flash+OTP:
Flash Programmer Settings	Execute Operation Help

Puc. 41

7. В окне «On-Chip Flash Programmer» нажать кнопку «Browse...».

8. B (рис. 42) выбрать файл появившемся окне «Example DAC.out» ИЗ папки «X:\DAC\DSP281x examples\DAC\Debug» И нажать кнопку «Открыть». Здесь X – условное обозначение литеры диска, содержащего исходные файлы для данного курса лабораторных работ.

Открыть				? ×
Папка: 🗀	Debug	•	← 🗈	📸 🎫
Example_C	AC.out			
Имя файла:	Example_DAC.out			Открыть
Тип файлов:	COFF Files (*.out)		•	Отмена

*Puc.* 42B окне «On-Chip Flash Programmer» нажать кнопку «Flash Programmer Settings...», что приведет к появлению окна «Flash Programmer Settings» (рис. 43).

Select DSP D F2810 F2811 F2812	evice to Program C F2801 C F2802 C F2806 C F2808	Options Load Symbols Display Tooltips Display Diagnostics	OK Cancel Help						
Select version of	Select version of Flash API Interface file:								

Puc. 43

9. В окне «Flash Programmer Settings» нажать кнопку «Browse...».

10. В появившемся окне (рис. 44) выбрать файл

«FlashAPIInterface 2812V2\_10.out» и нажать кнопку «Открыть».

Открыть			<u>?×</u>
Папка:		▼ ← 🖻	
🔲 FlashAPIIn	terface2812V2_10.out		
Имя файла:	FlashAPIInterface2812V2_10.out		Открыть
Тип файлов:	COFF Files (*.out)	•	Отмена

Puc. 44

11. В окне «Flash Programmer Settings» нажать кнопку «OK».

12. В окне «On-Chip Flash Programmer» нажать кнопку «Execute Operation», что приведет к запуску операций программирования микропроцессора.

ВНИМАНИЕ! ЗАПРЕЩАЕТСЯ ПОДАВАТЬ КАКИЕ-ЛИБО СИГНАЛЫ НА ВХОД СТЕНДА ЦОС ВО ВРЕМЯ ПРОГРАММИРОВАНИЯ МИКРОПРОЦЕССОРА

ВНИМАНИЕ! ЗАПРЕЩАЕТСЯ ОТКЛЮЧАТЬ ПИТАНИЕ СТЕНДА ЦОС ВО ВРЕМЯ ПРОГРАММИРОВАНИЯ МИКРОПРОЦЕССОРА

Микропроцессор считать успешно запрограммированным при отсутствии сообщений об ошибках в процессе программирования. После успешного завершения процедуры программирования следует закрыть окно «On-Chip Flash Programmer».

13. К разъему «Выход» Стенда ЦОС подключить осциллограф.

14. В меню главного окна среды Code Composer Studio 3.3 последовательно выбрать пункты «Debug»— «Run».

С помощью осциллографа убедиться в наличии последовательности пилообразных импульсов амплитудой (4,0±0,1) В и периодом (400±10) мс на выходе стенда ЦОС. Пример наблюдаемой осциллограммы показан на рис. 45

1.



Puc. 45

2. В меню главного окна среды Code Composer Studio 3.3 последовательно выбрать пункты «Debug»→«Halt».

3. С помощью ЦАП стенда ЦОС сформировать гармонический сигнал положительной полярности, имеющий размах 4,0 В и период 10,0 мс, период дискретизации сигнала 0,1 мс. Для этого внести соответствующие изменения в содержание файла «Example\_281xSpi\_FFDLB.c».

4. Выполнить действия п. 6–14.

5. Выполнить действия п. 16.

6. С помощью осциллографа убедиться в наличии гармонического сигнала с параметрами, указанными в п. 19 на выходе стенда ЦОС.

Контрольные вопросы

Приведите последовательность команд на языке C++, необходимую для установки параметров обмена информацией по интерфейсу SPI для микропроцессора TMS320F2812. Поясните назначение каждой приведенной команды.

1. Приведите последовательность команд на языке C++, необходимую для формирования гармонического сигнала с частотой 10 Гц и амплитудой 3 В с помощью цифроаналогового преобразователя стенда ЦОС (период дискретизации сигнала должен составлять T = 1 мс). Поясните назначение каждой приведенной команды.

2. Изложите принципы организации передачи данных по интерфейсу SPI.

3. Какие сигналы используются для обмена данными между устройствамипо интерфейсу SPI?

4. Поясните назначение каждого бита регистров интерфейса SPI микропроцессора TMS320F2812.

5. Приведите пример команды на языке C++, при выполнении которой происходит формирование импульсов синхросигнала интерфейса SPI.

6. Что такое синхронный интерфейс и асинхронный интерфейс? Приведите примеры синхронных и асинхронных интерфейсов.

7. Назовите максимальное значение частоты синхросигнала интерфейса

SPI.

8. Какое максимальное значение частоты может иметь синхросигнал ин-

терфейса SPI, вырабатываемый микропроцессором TMS320F2812?

9. Изложите принцип работы цифроаналогового преобразователя.

Как можно определить количество данных в буфере приемника интерфейса SPI микропроцессора TMS320F2812?

#### ЛАБОРАТОРНАЯ РАБОТА № 3

#### РАЗРАБОТКА НЕРЕКУРСИВНОГО ЦИФРОВОГО ФИЛЬТРА

Цель работы

Целью работы является получение навыков разработки ПО, реализующего нерекурсивные цифровые фильтры на основе процессора TMS320F2812.

Задание к работе

1. Спроектировать цифровой нерекурсивный фильтр.

2. Разработать ПО, реализующее спроектированный цифровой фильтр набазе стенда ЦОС.

3. Выполнить фильтрацию сигналов с использованием разработанногоцифрового фильтра.

4. Сделать выводы по результатам работы.

Пояснения к работе

Алгоритм линейной цифровой фильтрации в частном случае описываетсявыражением

гдежет принимать только целые значения 0, 1, 2, 3...;  $b_k$  отсчеты решетчатых функций. Целое число L – константа, которая определяет количество используемых отсчетов входного сигнала.

Пол проектированием (или синтезом) нерекурсивного цифрового фильтра понимается выбор такого набора коэффициентов  $\{b_k\},\ при котором характеристики получающегося фильтра$ удовлетворяют заданным требованиям [13, 14]. В настоящее время большое представлено количество математических программ. имеющих функции проектирования цифровых фильтров и расчета их характеристик. Одной из часто применяемых программ является пакет FDATool (Filter Design & Analysis Tool) программной среды Matlab [15]. Данная программа представляет собой очень мощное средство проектирования различных типов классических цифровых фильтров. В частности, она позволяет использовать различные проектирования фильтров методы И имеет возможность спроектированные фильтры. Кроме моделировать того, пакет FDATool производить экспорт результатов позволяет программы. проектирования фильтров Важнейшим В другие достоинством данной программы является возможность сохранения результатов проектирования фильтров (т. е. массива коэффициентов фильтра {  $b_k$  }) в виде заголовочного файла для ПО на языке C++.

Данный файл может быть использован для разработки ПО микропроцессоров. Иными словами, результаты проектирования цифрового фильтра в программной среде Matlab могут быть достаточно просто использованы при написании ПО для микропроцессоров (т. е. могут быть реализованы «в железе»).

Пакет синтеза и анализа фильтров FDATool вызывается из командной строки программной среды Matlab следующей командой:

>> fdatool

Пакет открывает окно (рис. 46) с множеством органов управления, предназначенных для выбора типа проектируемого цифрового фильтра и его параметров, а также для выбора средств визуализации результатов проектирования.

Const Flam totomicion     Flam totomicion	Filter Design & Analysis Tool - [ur File Edit Analysis Targets Window	titled.fda] Help		80
Control Tiles Information     The Social Status     The Socia	D 📽 🖬 🖨 🖪 💿 🖂 🖇	0 10 10 10 10 10	# 🗊 🖛 📾 🖬 👘 🕷	47
Enuties Directions The Order S0 Statem 1 Enuties Directions              Figure (60)               Impact A frequency (100)            Index Visions              Figure (100)               Figure (100)               Figure (100)            Index Visions              Figure (100)               Figure (100)               Figure (100)               Figure (100)            Index Visions               Figure (100)               Figure (100)               Figure (100)               Figure (100)            Interview              Figure (100)               Figure (100)               Figure (100)               Asset: (1            Interview              Outron:               Figure (120)               Figure (100)               Asset: (1            Interview              Outron:             Figure (120)               Figure (100)               Asset: (1	Current Filter Information	Filter Specifications		
File Type         File Order         File Order         Maphade Specification         Maphade Specification           C Revolution         Specify order         Specify order         Urder         Urd	Structure: Direct form FIR Dirder: 50 Soction: 1 Stable: Yes Source: Designed	• Mag. (dB)	Fpess Fstop	Fs/2 f(Hz)
R PR Equipple	Filter Type      Filter Type      Loopass     Bandatas     Bandatas     C Dotteendator      Design Netted      Ils Duterworth      R [Equipple]	Piter Order     Specily actor: 10     Minimum ocder     Options     Density factor: 16	Frequency Specification           Unit:         Hz           Fz:         45000           Ppess:         9600           Fstop:         12000	Apense: 1 Adaptive Specifications

Puc. 46

На рис. 46 показан вид окна при выборе проектирования ФНЧ. В окне просмотра представлены зоны допусков АЧХ с указанием их названий. После задания типа фильтра и его параметров достаточно нажать кнопку **Design Filter**-

для того, чтобы запустить программное конструирование

фильтра под заданные параметры. После этого, используя кнопки панели инструментов, можно посмотреть и, если надо, скорректировать полученные характеристики фильтра.

Пример:

Синтезировать нерекурсивный ФНЧ 9-го порядка с равномерными пульсациями АЧХ (equiripple) [16]. Частота среза составляет 50 Гц, частота дискретизации 1 кГц. Начало полосы задерживания 70 Гц. Результаты проектирования сохранить в виде заголовочного файла для ПО на языке С++.

На панели **Filter Туре** выберем тип фильтра, в нашем случае ФНЧ («Lowpass»). На панели **Design Method** выберем нерекурсивный фильтр («FIR») и аппроксимацию АЧХ «Equiripple». На панели **Filter Order** в поле

«Specify order» зададим порядок фильтра, равный 9. На панели **Frequency specification** в поле «Units» выберем единицы измерения частоты (Гц), в поле

«Fs» зададим частоту дискретизации (1000 Гц), в поле «Fstop» – значение границы полосы задерживания (70 Гц), в поле «Fpass» – частоту среза (50 Гц). Далее для начала процедуры проектирования необходимо нажать кнопку «Design Filter». Общий вид окна после перечисленных действий показан на рис. 47.



*Puc.* 47

Для экспорта коэффициентов синтезированного фильтра можно воспользоваться экспортом значений коэффициентов в заголовочный файл C/C++. В меню программы FDATool необходимо последовательно выбрать пункты

«File»— «Export to C Header File». Это приведет к появлению окна «Export to C Header File» (рис. 48).

📣 Export to C Header File				
<sub>C</sub> Variable names in C header file	,			
Numerato B	Numerator length: BL			
Number of sections: NS	j l			
Data type to use in export				
Export suggested	Double-precision			
	floating point			
C Export as:	Signed 32-bit integer 💌			
Fractional length: 31				
OK Cancel Help Apply				

쇠 Export to C Header File	
<sub>C</sub> Variable names in C header file	,
Numerato B	Numerator length: BL
Number of sections: NS	1
Data type to use in export	
C Export suggested:	Double-precision
	floating point
Export as:	Single-precision float
	Fractional length:
OKCance	Help Apply

*Puc.* 48

Puc. 49

В представленном окне на панели Data type to use in export следует отметить пункт «Export as:», в выпадающем списке выбрать пункт «Single-precision float» (рис. 49) и нажать кнопку «ОК». Далее необходимо сохранить созданный заголовочный файл на диске. Сохраненный файл может быть использован как заголовочный файл языке С/С++. Текст заголовочного для проектов на файла представлен в приложении Д.

Структурная схема спроектированного цифрового фильтра показана на рис. 50 (схема сформирована пакетом FDATool). Таким образом, синтезированный цифровой фильтр имеет прямую форму реализации [13].

Ход работы

Выполнить действия п. 1. 1 - 18«Подготовка главы рабочего места к выполнению лабораторных работ».

Запустить программную среду Matlab, в командной 2. строке средыMatlab выполнить команду:

>> fdatool

3. Синтезировать цифрового фильтра молель в FDATool согласно варианту заданий, указанному программе преподавателем. Получить графики характеристик синтезированного фильтра.



Puc. 50

Сохранить коэффициенты синтезированного цифрового фильтра в виде заголовочного файла для C/C++ с именем «filter\_coef.h».

4. Скопировать файл «filter\_coef.h» в папку

«X:\Dig\_filter1\DSP281x\_ examples\filter1», где X – условное обозначение литерыдиска, содержащего исходные файлы для данного курса лабораторных работ.

5. Скопировать файл «tmwtypes.h» из папки

«L:\MATLAB6p5\extern\ include», где L – условное обозначение литеры диска с установленной средой Matlab в папку «X:\Dig\_filter1\DSP281x\_examples\filter1».

6. Открыть проект ПО лабораторной работы № 3. Для этого в меню главного окна среды Code Composer Studio 3.3 последовательно выбрать пункты

«Project»  $\rightarrow$  «Open...» (рис. 51).

Project Open			? 🗙 🤉
Папка: 🗀	filter1	- ÷ È	r 🗐 🕂
Debug Example_fi Example_fi	lter1.CS_ lter1.pjt		
Имя файла:	Example_filter1.pjt		Открыть
Тип файлов:	Project Files (*.pjt)	•	Отмена
			Справка

Puc. 51

7. В появившемся окне выбрать файл «Example\_filter1.pjt» из папки

«X:\Dig\_filter1\DSP281x\_examples\filter1» и нажать кнопку «Открыть». При этом в левой части главного окна среды Code Composer Studio 3.3 появится

«дерево файлов» проекта, отображающее его структуру (рис. 52) [7].

8. Перемещаясь по «дереву файлов», выбрать двойным щелчком «мыши» файл «Example\_281xSpi\_FFDLB.c» (рис. 53). Вид окна после перечисленных действий представлен на рис. 54. Полный текст подпрограммы

«Example\_281xSpi\_FFDLB.с» для лабораторной работы №3 представлен в приложении Е.

9. В тексте файла «Example\_281xSpi\_FFDLB.c»

#### поместить следующую директиву [17]:

🤹 /F2812 eZdsp/cpu_0 - TM532	ОС28жж - Code Composer Studio - Not Connected	- 2 ×
File Edit View Project Debug	GEL Option Profile Tools DSP/BIOS Window Help	
🎦 🚅 🖬 🕹 🏝 🛍 🗠	- <b></b>	<b>€</b>   0°
<u> </u>	- ● 四 田 丞 ● ●	
R & <b>985</b> 80	E 4 8	
7) Files	Disassembly	
🕞 🥻 🚍 GEL files	◆BFFC00 28AD MOV GSP, 1^	
Projects	3FFC02 5616 CLBC AMODI	
7)	3FFC04 561A SETC MOM12	
	3FFC05 2940 CLRC PAGE	
	3FFC06 761F MOVW DP,#(	
	SEFCIA SD7F SUB ACC.	
MALTED	For Help, press F1	

Puc. 52



*Puc.* 53



Puc. 54

Выполнить компиляцию проекта. Для этого в главном окне среды Code Composer Studio 3.3 нажать кнопку В группе Соде Сотрование Собщение считать успешно откомпилированным при отсутствии сообщений об ошибках впроцессе компиляции.

10. Запрограммировать микропроцессор. Для этого в меню главного окна среды Code Composer Studio 3.3 последовательно выбрать пункты «Tools»→

«F28xx On-Chip Flash Programmer». При этом на экран будет выведено окно

«On-Chip Flash Programmer» (рис. 55).

In-Chip Flash Programmer	X
Clock Configuration OSOCLK (Mhz) 30.000 CLKINOV (/2)  PLLCR Value: 10 SYSCLKOUT (MHz): 150.0000	Exer Sector Selection
Code Security Planmoid           Kmy 7 (buck7)           Kmy 7 (buck7)           Kmy 6 (buck6)           Kmy 6 (buck6)           Kmy 7 (buck7)           Kmy 7 (buck7)<	Grandow Version     Construction     Construction
Flash Programmer Settings	Execute Operation Help

Puc. 55

11. В окне «On-Chip Flash Programmer» нажать кнопку «Browse...».

12.В появившемся окне (рис. 56) выбрать файл«Example\_filter1.out»из«X:\Dig\_filter1\DSP281x\_examples\filter1\Debug» и нажать кнопку«Открыть».

Открыть			<u>? ×</u>
Папка: 🗀	Debug	1	
Example_f	ilter1.out		
Имя файла:	Example_filter1.out		Открыть
Тип файлов:	COFF Files (*.out)	•	Отмена

Puc. 56

В окне «On-Chip Flash Programmer» нажать кнопку «Flash Programmer Settings...», что приведет к появлению окна «Flash Programmer Settings» (рис. 57).

Flash Programmer Settings					
Select DSP Device to Program F2810 F2801 F2811 F2802 F2812 F2806 F2808	Options Load Symbols Display Tooltips Display Diagnostics	OK Cancel Help			
Select version of Flash API Interface file:           C:\CCStudio_v3.3\plugins\Flash28xx\Algorithms\2812\FlashAPIInterface2812V2         Browse					

*Puc.* 57

13. В окне «Flash Programmer Settings» нажать кнопку «Browse...».

14. В появившемся окне (рис. 58) выбрать файл

«FlashAPIInterface2812 V2\_10.out» и нажать кнопку «Открыть».

Открыть			<u>? ×</u>
Папка:		▼ ← <b>€</b>	📸 🎟 -
FlashAPIIr	iterface2812V2_10.out		
Имя файла:	FlashAPIInterface2812V2_10.out		Открыть
Тип файлов:	COFF Files (*.out)	•	Отмена

Puc. 58

15. В окне «Flash Programmer Settings» нажать кнопку «OK»

16. В окне «On-Chip Flash Programmer» нажать кнопку «Execute Operation», что приведет к запуску операций программирования микропроцессора.

ВНИМАНИЕ! ЗАПРЕЩАЕТСЯ ПОДАВАТЬ КАКИЕ-ЛИБО СИГНАЛЫ НА

ВХОД СТЕНДА ЦОС ВО ВРЕМЯ ПРОГРАММИРОВАНИЯ МИКРОПРОЦЕССОРА

ВНИМАНИЕ! ЗАПРЕЩАЕТСЯ ОТКЛЮЧАТЬ ПИТАНИЕ СТЕНДА ЦОС

ВО ВРЕМЯ ПРОГРАММИРОВАНИЯ МИКРОПРОЦЕССОРА

Микропроцессор считать успешно запрограммированным при отсутствии сообщений об ошибках в процессе программирования. После успешного завершения процедуры программирования следует закрыть окно «On-Chip Flash Programmer».

17. К разъему «Вход» стенда ЦОС подключить генератор сигналов.К разъему «Выход» подключить осциллограф.

18. С генератора подать гармонический сигнал положительной полярностис частотой 1 Гц и размахом 2,5 В.

19. В меню главного окна среды Code Composer Studio 3.3 последовательно выбрать пункты «Debug»— «Run».

20. Изменяя частоту гармонического сигнала в диапазоне от 1 до 500 Гц (не менее 10 значений), с помощью осциллографа измерить амплитудные значения напряжения на выходе стенда ЦОС

и занести их в табл. 2.

21. По данным табл. 2 сделать заключение о соответствии АЧХ стенда ЦОС требованиям варианта задания (табл. 3), выбранного при выполнении п. 3.

Таблица 2

N⁰	Частота f,	Показания	осциллографа
	Гц	Um,	
	,	В	

#### Таблица З

#### Варианты заданий к лабораторной работе № 3

	N⁰		
	1	2	3
Тип	Нерекурсивный	Нерекурсивный	Нерекурсивный
алгоритма			
Тип	Равномерн	Фильтр	Равномерны
аппроксимацииАЧХ	ые пульсации	с	е пульсации
	(Equiripple)	произвольной	(Equiripple)
		амплитудой	
		(Least Pth Norm)	
Тип фильтра	ФНЧ	ФНЧ	ФНЧ
Порядок	9	9	9
фильтра			
Частота среза,	50	200	100
Гц			
Граничная	70	230	150
частота полосы			
задерживания, Гц			
f <sub>д</sub> , Гц	1000	1000	1000

Контрольные вопросы

1. Приведите последовательность команд на языке C++, необходимую для реализации нерекурсивного цифрового фильтра на базе стенда ЦОС. Поясните назначение каждой приведенной команды.

2. Перечислите основные характеристики цифровых фильтров.

3. Назовите и кратко опишите основные методы проектирования цифровых фильтров.

4. Как оформить результаты проектирования цифрового фильтра в программе FDATool, чтобы их можно было использовать для написания программ на языке C++?

5. Что такое импульсная характеристика цифрового фильтра?

6. Какая информация заложена в расположении нулей и полюсов функции передачи цифрового фильтра на комплексной плоскости?

7. Какие математические операции используются в алгоритме цифрового нерекурсивного фильтра?

8. Назовите основные формы реализации цифровых фильтров.

9. Какими характеристиками обладает цифровой фильтр, выполняющийалгоритм «скользящего среднего»?

10. Назовите основные способы описания цифровых фильтров.

#### ЛАБОРАТОРНАЯ РАБОТА № 4

# РАЗРАБОТКА РЕКУРСИВНОГО ЦИФРОВОГО ФИЛЬТРА

Цель работы

Целью работы является получение навыков разработки ПО, реализующего рекурсивные цифровые фильтры на основе процессора TMS320F2812.

Задание к работе

1. Спроектировать цифровой рекурсивный фильтр.

2. Разработать ПО, реализующее спроектированный цифровой фильтр набазе стенда ЦОС.

3. Выполнить фильтрацию сигналов с использованием разработанногоцифрового фильтра.

4. Сделать выводы по результатам работы.

Пояснения к работе

Рассмотренный ранее пакет FDATool также позволяет проектировать рекурсивные цифровые фильтры и моделировать их характеристики [15]. Как и в случае проектирования нерекурсивного цифрового фильтра, результаты синтезамогут быть экспортированы в другие программы в виде заголовочного файла для ПО на языке C++.

Алгоритм линейной цифровой фильтрации описывается выражением [12]

В результате проектирования рекурсивного цифрового фильтра создаются два массива коэффициентов {  $a_m$  } и {  $b_k$  } [13, 14]. При этом пакет FDATool выдает сведения о форме реализации

цифрового фильтра, т. е. отображается тип звеньев, из которых состоит цифровой фильтр, и тип соединения звеньев [13].

Пример:

Синтезировать рекурсивный ФНЧ, имеющий минимальный порядок с эллиптической функцией аппроксимации АЧХ (elliptic) [16]. Частота среза составляет 50 Гц, частота дискретизации 1 кГц. Начало полосы задерживания 70 Гц. Результаты проектирования сохранить в виде заголовочного файла для ПО на языке C++.

На панели **Filter Type** пакета FDATool выберем тип фильтра, в нашем случае ФНЧ («Lowpass»). На панели **Design Method** выберем нерекурсивный фильтр («IIR») и аппроксимацию АЧХ «Elliptic». На панели **Filter Order** отметим пункт «Minimum order». На панели **Frequency specification** в поле «Units» выберем единицы измерения частоты (Гц), в поле «Fs» зададим частоту дискретизации (1000 Гц), в поле «Fstop» – значение границы полосы задерживания (70 Гц), в поле «Fpass» – частоту среза (50 Гц). Далее для начала процедуры проектирования необходимо нажать кнопку «Design Filter». Общий вид окна после перечисленных действий показан на рис. 59.



*Puc.* 59

Структурная схема спроектированного цифрового фильтра

показана на рис. 60 (схема сформирована пакетом FDATool). Таким образом, синтезированный цифровой фильтр состоит из четырех звеньев канонической формы [13], последовательно соединенных между собой.

Для экспорта коэффициентов синтезированного фильтра также можно воспользоваться функцией создания заголовочного файла на языке C/C++. Текст заголовочного файла, созданного по результатам проектирования цифрового фильтра из рассмотренного примера, представлен в приложении Ж.



Puc. 60

Ход работы

1. Выполнить действия п. 1–18 главы «Подготовка рабочего места к выполнению лабораторных работ».

2. Запустить программную среду Matlab, в командной строке среды Matlab выполнить команду:

>> fdatool

3. Синтезировать модель цифрового фильтра в программе FDATool согласно варианту заданий, указанному преподавателем. Получить графики характеристик синтезированного фильтра.

4. Сохранить коэффициенты синтезированного цифрового фильтра в виде заголовочного файла для C/C++ с именем «filter\_coef2.h».

5. Скопировать файл «filter\_coef2.h» в папку

«X:\Dig\_filter2\DSP281x\_ examples\filter2», где X – условное обозначение литеры диска, содержащего исходные файлы для данного курса лабораторных работ.

6. Скопировать файл «tmwtypes.h» из папки

«L:\MATLAB6p5\extern\ include», где L – условное обозначение литеры диска с установленной средой Matlab, в папку «X:\Dig\_filter2\DSP281x\_examples\filter2».

7. Открыть проект ПО лабораторной работы №4. Для этого в меню главного окна среды Code Composer Studio 3.3 последовательно выбрать пункты

«Project»  $\rightarrow$  «Open...» (рис. 61).

Project Open			? 🗙
Папка: 🗀	filter2	• =	💣 🎟 -
Debug Example_f	ilter2.C5_ ilter2.pjt		
Имя файла:	Example_filter2.pjt		Открыть
Тип файлов:	Project Files (*.pjt)	•	Отмена
			Справка

Puc. 61

8. В появившемся окне выбрать файл «Example\_filter2.pjt» из папки

«X:\Dig\_filter2\DSP281x\_examples\filter2» и нажать кнопку «Открыть». При этом в левой части главного окна среды Code Composer Studio 3.3 появится

«дерево файлов» проекта, отображающее его структуру (рис. 62) [7].

9. Перемещаясь по «дереву файлов» выбрать двойным щелчком «мыши» файл «Example\_281xSpi\_FFDLB.c» (рис. 63). Вид окна после перечисленных действий представлен на рис. 64. Полный текст подпрограммы

«Example\_ 281хSpi\_FFDLB.с» для лабораторной работы № 4 представленв приложении 3.



Puc. 62



*Puc.* 63



Puc. 64

В тексте файла «Example\_281xSpi\_FFDLB.c» поместить следующую директиву [17]:

#include "filter\_coef2.h"

10. Выполнить компиляцию проекта. Для этого в главном окне среды Code Composer Studio 3.3 нажать кнопку В группе 🐼 🕮 📇 . Проект считать успешно откомпилированным при отсутствии сообщений об ошибках впроцессе компиляции.

11. Запрограммировать микропроцессор. Для этого в меню главного окна среды Code Composer Studio 3.3 последовательно

выбрать пункты «Tools»→

«F28xx On-Chip Flash Programmer». При этом на экран будет выведено окно

«On-Chip Flash Programmer» (рис. 65).

On-Chip Flash Programmer	×
Clock Configuration	Erase Sector Selection
OSCCLK (Mhz): 30.000	Sector A: (3F6000-3F7FF) Sector F: (3E4000-3E7FFF)
	Sector B: (3F4000-3F5FFF) Sector G: (3E0000-3E3FFF)
PLICE Volum	I Sector C: (3F0000-3F3FFF) I Sector H: (3DC000-3DFFFF)
	M Sector D: (3ELUUU-3EFFFF) M Sector I: (3DAUU-3DBFFF)
SYSCEKUUT (MHz): 150.0000	V SECIOLE: (SECOND-SEBLIC) V SECIOLS: (SPONDO-SPSCIC)
Code Security Password	Operation
Key 7 (0xAE7): FFFF	Please specify the COFF file to Program/Verify:
Key 6 (0xAE6): FFFF	CA Browse
Key 5 (0xAE5): FFFF	Erase, Program, Verify C Depletion Recovery
Key 4 (0xAE4): FFFF	C Erase Only C Frequency Test
Key 3 (0xAE3): FFFF	C Program, Verify Register: GPAMux
Key 2 (0xAE2): FFFF	C Verifu Only Pin: PWM1 (0)
Key 1 (0xAE1): FFFF	Each Deader Mark Chater 15 C. C. Coloniste Checkerme
Key 0 (0xAE0): FFFF	
Itelack Leak	Flash Page Wait State: 15 Flash:
	OTP Wait State: 31 💌 OTP:
Program Password	C Load RAM Only Flash+0TP:
Flash Programmer Settings	Execute Operation Help

Puc. 65

12. В окне «On-Chip Flash Programmer» нажать кнопку «Browse...».

13. В появившемся окне (рис. 66) выбрать файл«Example\_filter2.out»из«X:\Dig\_filter2\DSP281x\_examples\filter2\Debug» и нажать кнопку«Открыть».

Открыть			<u>?</u> ×
Папка: 🗀 De	ebug 💌 🗲	* 🖻	
Example_filte	2.out		
Имя файла: Е	xample_filter2.out		Открыть
Тип файлов: 🛛	OFF Files (*.out)	-	Отмена

Puc. 66

14. В окне «On-Chip Flash Programmer» нажать кнопку «Flash Programmer Settings...», что приведет к появлению окна

#### «Flash Programmer Settings» (рис. 67).

Flash Programmer Settings	0 f			
C F2810 C F2801 C F2811 C F2802 C F2812 C F2802 C F2808		OK Cancel Help		
Select version of Flash API Interface file: C:\CCStudio_v3.3\plugins\Flash28xx\Algorithms\2812\FlashAPIInterface2812V2 Browse				

Puc. 67

15. В окне «Flash Programmer Settings» нажать кнопку «Browse...».

16. В появившемся окне (рис. 68) выбрать файл

«FlashAPIInterface2812V2\_ 10.out» и нажать кнопку «Открыть».

Открыть			<u>? ×</u>
Папка:		• <del>•</del> •	
FlashAPIIn	terface2812V2_10.out		
Имя файла:	FlashAPIInterface2812V2_10.out		Открыть
Тип файлов:	COFF Files (*.out)	•	Отмена

Puc. 68

17. В окне «Flash Programmer Settings» нажать кнопку «OK».

Микропроцессор считать успешно запрограммированным при отсутствии сообщений об ошибках в процессе программирования. После успешного завершения процедуры программирования следует закрыть окно «On-Chip Flash Programmer».

18. К разъему «Вход» стенда ЦОС подключить

генератор сигналов.К разъему «Выход» подключить осциллограф.

19. С генератора подать гармонический сигнал положительной полярностис частотой 1 Гц и размахом 2,5 В.

20. В меню главного окна среды Code Composer Studio 3.3 последовательно выбрать пункты «Debug»→«Run».

21. Изменяя частоту гармонического сигнала в диапазоне от 1 до 500 Гц (не менее 10 значений), с помощью осциллографа измерить амплитудные значения напряжения на выходе стенда ЦОС и занести их в табл. 4.

22. По данным табл. 4 сделать заключение о соответствии АЧХ стенда ЦОС требованиям варианта задания (табл. 5), выбранного при выполнении п. 3.

# Таблица 4

N⁰	Частота f,	Показания осциллографа Um,
	Гц	В

# Таблица 5

#### Варианты заданий к лабораторной работе № 4

	N⁰		
	1	2	3
Тип	Рекурсивный	Рекурсивный	Рекурсивный
алгоритма			
1	2	3	4
Тип	Эллиптическая	Эллиптическая	Эллиптическая
аппроксимации			
АЧХ	функция аппрок-	функция аппрокси-	функция аппрок-
	симации АЧХ	мации АЧХ	симации АЧХ
	(Elliptic)	(Elliptic)	(Elliptic)

#### Окончание табл. 5

1	2	3	4	
Тип фильтра	ФНЧ	ФНЧ	ФНЧ	
Порядок фильтра	8	8	8	
Частота среза, Гц	50	10	100	
Затухание в полосе пропускания, дБ	1	1	1	
Затухание в полосе задерживания	80	80	80	
f <sub>д</sub> , Гц	1000	1000	1000	

1. Приведите последовательность команд на языке C++, необходимую для реализации рекурсивного цифрового фильтра на базе стенда ЦОС. Поясните назначение каждой приведенной команды.

2. Перечислите известные вам типы аппроксимации АЧХ рекурсивных дискретных фильтров.

3. Что такое устойчивость и реализуемость дискретных фильтров?

4. Как оценивается устойчивость цифрового рекурсивного фильтра на основе расположения нулей и полюсов функции передачи на комплексной плоскости?

5. Назовите и кратко опишите формы реализации и возможные соединениядискретных фильтров.

6. Какие особенности имеет импульсная характеристика цифрового рекурсивного фильтра?

7. Какие математические операции используются в алгоритме цифрового рекурсивного фильтра?

8. Какие требования предъявляются к фазочастотной характеристике цифрового фильтра?

9. Что такое групповая задержка дискретного фильтра?

10. Как влияет квантование коэффициентов на характеристики цифровых фильтров?

Заключение

Представленные лабораторные работы позволяют изучить вопросы реализации алгоритмов обработки сигналов на основе цифровых сигнальных процессоров. В ходе разработки ПО и работы с аппаратной частью микропроцессора совершенствуются навыки создания программ на языке C++. Реализация алгоритмов обработки сигналов дает возможность ознакомиться с работой типовых периферийных устройств, входящих в состав микросхем цифровых сигнальных процессоров.

Подробное описание интерфейса среды Code Composer Studio позволяет освоить различные стадии разработки ПО (от начала создания нового проекта до окончательной отладки программ и Flash-памяти процессора). программирования Взаимодействие используемых разработки ПО сред И математического наглядно представляет реализацию результатов моделирования математическихрасчетов в физическом устройстве.

Выполнение цикла лабораторных работ значительно упрощает изучение других сигнальных процессоров с более сложной архитектурой.

#### Список литературы

1. TMS320x281x DSP. Analog-to-Digital Converter (ADC). Reference Guide(SPRU060D). Texas Instruments, 2005.

2. eZdsp F2812 Technical Reference. Texas Instruments, 2000.

3. Сергиенко А.Б. Цифровая обработка сигналов / А.Б. Сергиенко. – СПб. :Питер, 2003. – 608 с.

4. TMS320F2810, TMS320F2811, TMS320F2812, TMS320C2810, TMS320C2811, TMS320C2812 Digital Signal Processors. Data Manual (SPRS174T). Texas Instruments, 2000.

5. Programming TMS320x28xx and 28xxx Peripherals in C/C++. ApplicationReport (SPRAA85C). Texas Instruments. 2012.

6. TMS320x281xDSPSystemControlandInterrupts.ReferenceGuide (SPRU078G).Texas Instruments, 2012.

7. Getting Started With TMS320C28x Digital Signal Controllers. ApplicationReport (SPRAAM0A). Texas Instruments, 2007.

8. Дьяконов В. П. МАТLAB 6.5 SP1/7 + Simulink 5/6 Основы применения /В. П. Дьяконов. – М. : СОЛОН-Пресс, 2005. – 800 с.

9. Гук М. Аппаратные интерфейсы ПК / М. Гук. – СПб. : Питер, 2002. –528 с.

10. TMS320x281x Serial Peripheral Interface. Reference Guide (SPRU059E).Texas Instruments, 2009.

12. Гольденберг Л. М. Цифровая обработка сигналов / Л. М. Гольденберг, Б. Д. Матюшкин, М. Н. Поляк. – М. : Радио и связь, 1990. – 256 с.

13. Сергиенко А. Б. Цифровая обработка сигналов / А. Б. Сергиенко. –3-е изд. – СПб. : БХВ-Петербург, 2011. – 768 с.

14. Айфичер Э. Цифровая обработка сигналов. Практический подход / Э. Айфичер, Б. Джервис. – М. : Вильямс, 2004. – 992 с.

15. Дьяконов В. П. МАТLAВ. Обработка сигналов и изображений : спец. справочник / В. П. Дьяконов. – СПб. : Питер,

2002. – 608 c.

16. Основы цифровой обработки сигналов : курс лекций / А. И. Солонина[и др.]. – СПб. : БХВ-Петербург, 2003. – 608 с.

TMS320C28x Optimizing C/C++ Compiler v6.1. User's Guide(SPRU514E). Texas Instruments, 2012.

12. Уидроу Б., Адаптивная обработка сигналов / Б. Уидроу, С. Стирнз. –М.: Радио и связь, 1989. – 440 с.

13. Методы робастного, нейро-нечеткого и адаптивного управления / К. А. Пупков [и др.]; под ред. Н. Д. Егупова. – М. : МГТУ им. Н. Э. Баумана, 2001. – 744 с.

14. Arakawa K., Arakawa Y. Digital signal processing using fuzzy clustering for nonstationary signal // Proceedings of international Fuzzy Engineering Symposium. Nov. 13 th -15 th, 1991. Yokohama, Japan: Fuzzy Engineering toward. Human Friendly Systems, vol.2. IFES-91. – P. 877–888.

15. Вешкурцев Ю. М. Цифровой фильтр на основе теории нечетких множеств с адаптивно изменяемыми функциями принадлежности / Ю. М. Вешкурцев., Е. Д. Бычков, Д. А. Титов // Известия вузов России. Радиоэлектроника. – 2007. – Вып. 2. – С. 43–50.

16. C281x C/C++ Header Files and Peripheral Examples (SPRC097). Texas Instruments, 2007.

# ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ УСТРОЙСТВ ОБРАБОТКИ СИГНАЛОВ

#### МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ по дисциплине «Проектирование цифровых устройств обработки сигналов» для студентов направления 12.03.01 «Проектирование и технология радиоэлектронных средств»(профиль «Проектирование и технология радиоэлектронных средств») всех форм обучения

Составитель: Ципина Наталья Викторовна

Компьютерный набор Н. В. Ципиной

Подписано к изданию\_\_\_\_. Уч.-изд. л.\_.

ФГБОУ ВО «Воронежский государственный технический университет» 394026 Воронеж, Московский просп., 14