

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Запорізький національний технічний університет

МІКРОПРОЦЕСОРИ

Методичні вказівки до лабораторних робіт з дисципліни «Цифрові пристрої та мікропроцесори» студентів фаху 7.090701 «радіотехніка»
усіх форм навчання

Частина 4 "Сигнальні процесори з фіксованою крапкою"

2009

МІКРОПРОЦЕСОРИ. Методичні вказівки до лабораторних робіт з дисципліни "Цифрові пристрої та мікропроцесори" для студентів фаху 7.090701 «радіотехніка» усіх форм навчання.

Частина 4 "Сигнальні процесори з фіксованою крапкою" / Укл. В.І. Мисленков, А.С. Сіренко - ЗНТУ. - Запоріжжя, 2009. – 42 с.

Укладачі: ст.. викладач В.І. Мисленков,
 Асистент А.С. Сіренко

Рецензент: ст.. викладач Ю.Л. Мейстер

Відповідальний
за випуск: зав. лабораторією В.П. Колесніков

Затверджено:
на засіданні кафедри
«радіотехніки та телекомунікацій»
Протокол № 12__ від 24 06__ 2009

ЗМІСТ

ВИМОГИ ДО ОФОРМЛЕННЯ ЗВІТУ З ЛАБОРАТОРНОЇ РОБОТИ	4
1 АРХІТЕКТУРА ADSP-2181.....	4
1.1 Завдання	4
1.2 Порядок виконання роботи	5
1.3 Контрольні питання	8
2 ЛАБОРАТОРНА РОБОТА № 2	9
АРИФМЕТИКА ЦПОС З ФІКСОВАНОЮ КРАПКОЮ.....	9
2.1 Завдання.....	9
2.2 Порядок виконання роботи	11
2.3 Контрольні питання	13
3 ЛАБОРАТОРНА РОБОТА № 3	14
УЧБОВИЙ СТЕНД ADDS2181 EZ-KIT LITE	14
3.1 Завдання.....	14
3.2 Порядок виконання роботи	14
3.3 Контрольні питання	17
4 ЛАБОРАТОРНА РОБОТА № 4	18
МОВА АСЕМБЛЕРА ЦПОС ADSP-2181	18
4.1 Завдання.....	18
4.2 Порядок виконання роботи	18
4.3 Контрольні питання	20
5 ЛАБОРАТОРНА РОБОТА № 5	21
МАТЕМАТИЧНІ ОБЧИСЛЕННЯ НА ЦПОС	21
ADSP-2181.....	21
5.1 Завдання.....	21
5.2 Порядок виконання роботи	21
5.3 Контрольні питання	25
6 ЛАБОРАТОРНА РОБОТА № 6	26
ПРОЕКТУВАННЯ FIR-ФІЛЬТРІВ НА ЦПОС.....	26
ADSP-2181.....	26
6.1 Завдання.....	26
6.2 Порядок виконання роботи	26
6.3 Контрольні питання	29
ЛІТЕРАТУРА	30
Додаток А_Мікропроцесор ADSP-2181	31
Додаток Б_Опис стенда ADDS-2181 EZ-KITLite.....	40

ВИМОГИ ДО ОФОРМЛЕННЯ ЗВІТУ З ЛАБОРАТОРНОЇ РОБОТИ

Звіт повинен містити:

- титульний лист із назвою теми, номером завдання та інших звичайних атрибутів;
- мету лабораторної роботи;
- завдання, видане на лабораторну роботу;
- матеріали, необхідні для представлення результатів роботи (текст програми, таблиці, ілюстрації);
- зміст контрольних запитань;
- матеріали, необхідні для представлення відповіді на контрольне запитання;
- висновки.

1 ЛАБОРАТОРНА РОБОТА № 1 АРХІТЕКТУРА ADSP-2181

Мета роботи: вивчити архітектуру і технічні характеристики ЦПОС ADSP-2181. Придбати практичні навички роботи з програмою ADSP-2181 Navigator. Ознайомитися з технічною документацією, що відноситься до сімейства ЦПОС ADSP-21xx.

1.1 Завдання

1.1.1 Вивчення пристрою генераторів адреси ЦПОС сімейства ADSP-21XX.

1.1.2 Вивчення принципу роботи блоків ALU і MAC сигнальних процесорів сімейства ADSP-21XX.

1.1.3 Вивчення принципу роботи блоку Shifter ЦПОС сімейства ADSP-21XX.

1.1.4 Вивчення пристрою контролера BDMA сигнальних процесорів сімейства ADSP-218x.

1.1.5 Вивчення принципу роботи порту IDMA ЦПОС сімейства ADSP-218x.

1.1.6 Вивчення пристрою програмного автомата ЦПОС сімейства ADSP-21XX.

1.2 Порядок виконання роботи

1.2.1 Вивчити пристрій генераторів адреси ЦПОС сімейства ADSP- 21XX.

- запустити програму ADSP-218x Navigator (файл 218xNavigator.exe);
- відкрити вікно [003] ;
- у вікнах [004] , [005] , [006] , [007] містяться необхідні теоретичні відомості;
- запустити приклад 1. Вивчити порядок формування адреси в циклічному буфері DAG1;
- у вікні [009] описані розходження в пристрої DAG1 і DAG2;
- самостійно розібрати приклад роботи генератора адреси при інших початкових значеннях, що зберігаються в регістрах L, I та M (завдання видає викладач). Результат роботи оформити у вигляді таблиці (табл. 1.1).

Таблиця 1.1 – Формування адреси в циклічному буфері

№ циклу	Регістр L	Регістр I	Регістр M

1.2.2 Вивчення принципу роботи блоків ALU і MAC сигнальних процесорів сімейства ADSP-21XX:

- запустити програму ADSP-218x Navigator (файл 218xNavigator.exe) ;
- відкрити вікно [017] ;
- у вікнах [018] і [019] містяться необхідні теоретичні відомості;
- запустити приклад 3. Вивчити порядок виконання операції додавання в ALU;
- відкрити вікно [021] ;
- у вікнах [022] і [023] містяться необхідні теоретичні відомості;

- запустити приклад 4. Вивчити порядок виконання операції множення-накопичення в MAC;
- запустити приклад 5. Вивчити порядок виконання операції множення-накопичення в MAC для випадку насичення;
- самостійно розібрати приклад роботи блоку MAC при виконанні частини програми, заданої викладачем. Результат роботи оформити у вигляді таблиці (табл. 1.2);

Таблиця 1.2 – Виконання операції множення-накопичення в блоці MAC

№ циклу	MX	MY	MR2	MR1	MR0	MF

1.2.3 Вивчення принципу роботи блоку Shifter ЦПОС сімейства ADSP-21XX.

- запустити програму ADSP-218x Navigator (файл 218xNavigator.exe);
- відкрити вікно [045] ;
- у вікнах [046] , [047] і [050] містяться необхідні теоретичні відомості;
- запустити приклад 10. Вивчити порядок виконання операції зрушення в блоці Shifter;
- відкрити вікно [049], що містить відомості про операцію визначення максимальної експоненти блоку даних;
- запустити приклади 11, 12;
- відкрити вікно [052], що містить відомості про операцію нормалізації даних;
- самостійно розібрати приклад роботи блоку Shifter при виконанні ділянки програми, заданої викладачем. Результат роботи оформити у вигляді таблиці (табл. 1.3).

Таблиця 1.3 – Виконання операцій у блоці Shifter

№ циклу	SI	SE	SB	SR1	SR0

1.2.4 вивчення пристрою контролера BDMA сигнальних процесорів сімейства ADSP-218x:

- запустити програму ADSP-218x Navigator (файл 218xNavigator.exe);
- відкрити вікно [060];
- у вікнах [060] , [064] , [065] , [066] і [067] містяться необхідні теоретичні відомості;
- запустити приклад 13. Вивчити порядок читання 8-розрядних даних, вирівняних методом LSB;
- запустити приклад 14. Вивчити порядок читання 8-розрядних даних, вирівняних методом MSB;
- запустити приклад 15. Вивчити порядок читання 16-розрядних даних;
- запустити приклад 16. Вивчити порядок читання 24-розрядних даних;
- порівняти чотири методи читання даних з використанням контролера BDMA. Результати порівняння звести в (табл. 1.4) ;

Таблиця 1.4 – Виконання операцій читання контролером BDMA

Метод читання даних	Особливості
8-Bit LSB Align Read	
8-Bit MSB Align Read	
16-Bit Read	
24-Bit Read	

- запустити приклад 17. Вивчити порядок запису 8-розрядних даних, вирівняних методом LSB;
- запустити приклад 14. Вивчити порядок запису 8-розрядних даних, вирівняних методом MSB;
- запустити приклад 15. Вивчити порядок запису 16-розрядних даних;
- запустити приклад 16. Вивчити порядок запису 24-розрядних даних;
- порівняти чотири методи запису даних з використанням контролера BDMA. Результати порівняння занести в (табл. 1.5).

Таблиця 1.5 – Виконання операцій запису контролером BDMA

Метод запису даних	Особливості
8-Bit LSB Align Write	
8-Bit MSB Align Write	
16-Bit Write	
24-Bit Write	

1.2.5 Вивчення принципу роботи порту IDMA ЦПОС сімейства ADSP-218x:

- запустити програму ADSP-218x Navigator (файл 218xNavigator.exe);
- відкрити вікно [118];
- у вікнах [119] , [120] і [121] містяться необхідні теоретичні відомості;
- запустити приклад 21. Вивчити порядок роботи ЦПОС разом з хост-контроллером;
- скласти схему протоколу обміну даними через порт IDMA.

1.2.6 Вивчення пристрою програмного автомата ЦПОС сімейства ADSP-21XX:

- запустити програму ADSP-218x Navigator (файл 218xNavigator.exe);
- відкрити вікно [027];
- у вікнах [028] , [029] , [030] , [031] , [032] , [033] та ін. містяться необхідні теоретичні відомості;
- запустити приклад 7. Вивчити принцип роботи програмного автомата;
- запустити приклад 6. Вивчити метод апаратної організації циклу в ЦПОС;
- скласти блок-схему програмного автомата ЦПОС і описати призначення його основних частин.

1.3 Контрольні питання

1.3.1 Області використання сигнальних процесорів.

1.3.2 Особливості архітектури сучасних ЦПОС.

1.3.3 Основні принципи роботи контролерів переривань і ПДП.

1.3.4 Технічні характеристики ЦПОС сімейства ADSP-218х.

1.3.5 Порівняльна характеристика різних класів мікропроцесорів.

1.3.6 Класифікація архітектури обчислювальних пристроїв.

1.3.7 Приклад використання ЦПОС у складі аналогового модему.

1.3.8 Приклад використання ЦПОС у складі мобільного телефону.

1.3.9 Приклад використання ЦПОС у складі звукової карти ПК;

1.3.10 Приклад використання ЦПОС у складі системи ЦОС;

1.3.11 Переваги і недоліки використання програмного автомата в складі сигнального процесора.

1.3.12 Архітектура сигнальних процесорів з фіксованою крапкою фірми Analog Devices.

2 ЛАБОРАТОРНА РОБОТА № 2 АРИФМЕТИКА ЦПОС З ФІКСОВАНОЮ КРАПКОЮ

Мета роботи: вивчити формати представлення чисел у ЦПОС ADSP-2181 і правила їх використання у різних додатках ЦОС. Придбати практичні навички, необхідні для проектування систем цифрової обробки сигналів. Ознайомитися з технічною документацією, що відноситься до наявних інструментальних програм.

2.1 Завдання

2.1.1 Програми додавання, віднімання і множення дробових чисел (подвійна точність).

2.1.2 Програми додавання, віднімання і множення дробових чисел (потрійна точність).

2.1.3 Програми множення цілих чисел (подвійна і потрійна точність).

2.1.4 Програма розподілу чисел одинарної точності без знаку.

2.1.5 Програма розподілу чисел одинарної точності зі знаком.

2.1.6 Програма мовою Сі для перетворення форматів представлення даних.

2.2 Порядок виконання роботи

2.2.1 Програми додавання, віднімання і множення дробових чисел (подвійна точність):

- одержати завдання у викладача. За допомогою калькулятора розрахувати очікуваний результат;
- створити на диску свій робочий каталог (наприклад, C:\Work\lab2). Скопіювати в нього файли asm21.exe, ld21.exe, spl21.exe;
- у каталозі \Sources знайти заготовки програм (файли dpa.dsp, dps.dsp і dpm.dsp). Скопіювати їх у свій робочий каталог;
- за допомогою текстового редактора змінити текст програм відповідно до завдання;
- за допомогою програм ASM21 і LD21 одержати модулі, що виконуються, (dpa.exe, dps.dsp і dpm.dsp) ;
- запустити програму Visual DSP Debugger. Налаштувати режим симуляції процесора ADSP-2181;
- завантажити в інтегроване середовище програму dpa.exe і виконати її покрокове налагодження. Записати зміни значень у регістрах ЦПОС ADSP-2181. Порівняти отриманий результат з очікуваним;
- завантажити в інтегроване середовище програму dps.exe і виконати її покрокове налагодження. Записати зміни значень у регістрах ЦПОС ADSP-2181. Порівняти отриманий результат з очікуваним;
- завантажити в інтегроване середовище програму dpm.exe і виконати її покрокове налагодження. Записати зміни значень у регістрах ЦПОС ADSP-2181. Порівняти отриманий результат з очікуваним;
- програми додавання, віднімання і множення дробових чисел (потрійна точність) ;
- одержати завдання у викладача. За допомогою калькулятора розрахувати очікуваний результат;

- створити на диску свій робочий каталог (наприклад, C:\Work\lab2). Скопіювати в нього файли asm21.exe, ld21.exe, spl21.exe;
- у каталозі \Sources знайти заготовки програм (файли tra.dsp, tps.dsp і tpm.dsp). Скопіювати їх у свій робочий каталог;
- за допомогою текстового редактора змінити текст програм відповідно до завдання;
- за допомогою програм ASM21 і LD21 одержати модулі, що виконуються, (tra.exe, tps.exe і tpm.exe) ;
- запустити програму Visual DSP Debugger. Налаштувати режим симуляції процесора ADSP-2181;
- завантажити в інтегроване середовище програму tra.exe і виконати її покрокове налагодження. Записати зміни значень у регістрах ЦПОС ADSP-2181. Порівняти отриманий результат з очікуваним;
- завантажити в інтегроване середовище програму tps.exe і виконати її покрокове налагодження. Записати зміни значень у регістрах ЦПОС ADSP-2181. Порівняти отриманий результат з очікуваним;
- завантажити в інтегроване середовище програму tpm.exe і виконати її покрокове налагодження. Записати зміни значень у регістрах ЦПОС ADSP-2181. Порівняти отриманий результат з очікуваним.

2.2.2 Програми множення цілих чисел (подвійна і потрійна точність):

- одержати завдання у викладача. За допомогою калькулятора розрахувати очікуваний результат;
- створити на диску свій робочий каталог (наприклад, C:\Work\lab2). Скопіювати в нього файли asm21.exe, ld21.exe, spl21.exe;
- у каталозі \Sources знайти заготовки програм (файли idpm.dsp і itpm.dsp). Скопіювати їх у свій робочий каталог;
- за допомогою текстового редактора змінити текст програм відповідно до завдання;
- за допомогою програм ASM21 і LD21 одержати модулі, що виконуються, (idpm.exe і itpm.exe) ;

- запустити програму Visual DSP Debugger. Налаштувати режим симуляції процесора ADSP-2181;
- Завантажити в інтегроване середовище програму idpm.exe і виконати її покрокове налагодження. Записати зміни значень у регістрах ЦПОС ADSP-2181. Порівняти отриманий результат з очікуваним;
- завантажити в інтегроване середовище програму itpm.exe і виконати її покрокове налагодження. Записати зміни значень у регістрах ЦПОС ADSP-2181. Порівняти отриманий результат з очікуваним.

2.2.3 Програма розподілу чисел одинарної точності (без знака):

- одержати завдання у викладача. За допомогою калькулятора розрахувати очікуваний результат;
- створити на диску свій робочий каталог (наприклад, C:\Work\lab2). Скопіювати в нього файли asm21.exe, ld21.exe, spl21.exe;
- у каталозі \Sources знайти заготовку програми (файл sdivq.dsp). Скопіювати її у свій робочий каталог;
- за допомогою текстового редактора змінити текст програми відповідно до завдання;
- за допомогою програм ASM21 і LD21 одержати модуль, що виконується, (sdivq.exe);
- запустити програму Visual DSP Debugger. Налаштувати режим симуляції процесора ADSP-2181;
- завантажити в інтегроване середовище програму sdivq.exe і виконати її покрокове налагодження. Записати зміни значень у регістрах ЦПОС ADSP-2181. Порівняти отриманий результат з очікуваним;
- за допомогою програми SPL21 розщепити програму sdivq.exe для запису в ППЗУ. Формат задається викладачем.

2.2.4 Програма розподілу чисел одинарної точності (зі знаком):

- одержати завдання у викладача. За допомогою калькулятора розрахувати очікуваний результат;
- створити на диску свій робочий каталог (наприклад, C:\Work\lab2). Скопіювати в нього файли asm21.exe, ld21.exe, spl21.exe;

- у каталозі \Sources знайти заготовку програми (файл sdivs.dsp). Скопіювати її у свій робочий каталог;
- за допомогою текстового редактора змінити текст програми відповідно до завдання;
- за допомогою програм ASM21 і LD21 одержати модуль, що виконується, (sdivs.exe) ;
- запустити програму Visual DSP Debugger. Налаштувати режим симуляції процесора ADSP-2181;
- завантажити в інтегроване середовище програму sdivs.exe і виконати її покрокове налагодження. Записати зміни значень у регістрах ЦПОС ADSP-2181. Порівняти отриманий результат з очікуваним;
- За допомогою програми SPL21 розщепити програму sdivs.exe для запису в ППЗУ. Формат задається викладачем.

2.2.6 Програма мовою Сі для перетворення форматів представлення даних. Завдання видає викладач.

2.3 Контрольні питання

2.3.1 Властивості додаткового коду представлення дробових і цілих чисел.

2.3.2 Формати 1.15 і 16.0.

2.3.3 Основні арифметичні операції, які можна виконати на ЦПОС сімейства ADSP-21XX.

2.3.4 Операція нормалізації дробових чисел. Як вона реалізована в сигнальному процесорі ADSP-2181.

2.3.5 Динамічний діапазон представлення чисел.

2.3.6 Призначення і використання програми ASM21.

2.3.7 Призначення і використання програми LD21.

2.3.8 Призначення і використання програми SPL21.

2.3.9 Етапи розробки програм для ЦПОС сімейства ADSP-21XX.

2.3.10 Структура програми мовою асемблера ADSP-21XX.

2.3.11 Особливості використання пам'яті даних і пам'яті програм ЦПОС ADSP-2181.

2.3.12 Порівняння можливостей програм SIM21 і Visual DSP Debugger.

3 ЛАБОРАТОРНА РОБОТА № 3

УЧБОВИЙ СТЕНД ADDS2181 EZ-KIT LITE

Мета роботи: вивчити пристрій і принцип дії учбового стенда ADDS2181 EZ-KIT Lite. Придбати практичні навички роботи з ним. Ознайомитися з методикою, що рекомендується виробником розробки систем обробки сигналів і методами налагодження програм, що рекомендує виробник.

3.1 Завдання

3.1.1 Налагодження програми мовою асемблера ADSP-2181.

3.1.2 Налагодження програми мовою Сі для ADSP-2181.

3.1.3 Програма мовою асемблера для налагодження в середовищі Visual DSP.

3.1.4 Програма мовою асемблера для стенда ADDS-2181 EZ-KIT Lite.

3.1.5 Розробка архітектурного файлу системи.

3.1.6 Підготовка файлу для прошивання ППЗУ.

3.2 Порядок виконання роботи

3.2.1 Налагодження програми мовою асемблера ADSP-2181:

- одержати завдання у викладача;
- створити на диску свій робочий каталог (наприклад, C:\Work\lab3);
- скопіювати в нього файли asm21.exe і ld21.exe;
- за допомогою текстового редактора скласти текст програм відповідно до завдання;
- скласти .BAT-файл пакетного виконання завдання;
- у пакетному режимі виконати завдання;
- за допомогою програми EZ-KIT Host завантажити модуль, що виконується, у пам'ять програм сигнального процесора і виконати його. Використовувати кнопку RESET для виконання апаратного скидання стану;

- за допомогою програми Visual DSP Debugger виконати покрокове виконання тієї ж самої програми. Скласти таблицю зміни значень у регістрах процесора;

- зміст командного файлу і таблиці зміни значень у регістрах процесора представити в звіті.

3.2.2 Налаштування програми мовою Cі для ADSP-2181:

- одержати завдання у викладача;
- створити на диску свій робочий каталог (наприклад, C:\Work\lab3);

- скопіювати в нього файли asm21.exe, ld21.exe, g21.exe;
- за допомогою текстового редактора скласти текст програм відповідно до завдання;

- скласти .BAT-файл пакетного виконання завдання;
- у пакетному режимі виконати завдання;
- за допомогою програми EZ-KIT Host завантажити модуль, що виконується, у пам'ять програм сигнального процесора і виконати його. Використовувати кнопку RESET для виконання апаратного скидання стану;

- за допомогою програми Visual DSP Debugger виконати покрокове виконання тієї ж самої програми. Скласти таблицю зміни значень у регістрах процесора;

- зміст командного файлу і таблиці зміни значень у регістрах процесора представити в звіті.

3.2.3 Програма мовою асемблера для налаштування в середовищі Visual DSP:

- одержати завдання у викладача;
- створити на диску свій робочий каталог (наприклад, C:\Work\lab3);

- скопіювати в нього файли asm21.exe і ld21.exe;
- за допомогою текстового редактора скласти текст програм відповідно до завдання;

- скласти .BAT-файл пакетного виконання завдання;
- у пакетному режимі виконати завдання;
- за допомогою програми EZ-KIT Host завантажити модуль, що виконується, у пам'ять програм сигнального процесора і

виконати його. Використовувати кнопку RESET для виконання апаратного скидання стану;

- за допомогою програми Visual DSP Debugger виконати покрокове виконання тієї ж самої програми. Скласти таблицю зміни значень у регістрах процесора;

- зміст командного файлу і таблиці зміни значень у регістрах процесора представити в звіті.

3.2.4 Програма мовою асемблера для стенда ADDS-2181 EZ-KIT Lite:

- одержати завдання у викладача;
- створити на диску свій робочий каталог (наприклад, C:\Work\lab3).

- скопіювати в нього файли asm21.exe і ld21.exe;
- за допомогою текстового редактора скласти текст програм відповідно до завдання;

- скласти .BAT-файл пакетного виконання завдання;
- у пакетному режимі виконати завдання;
- за допомогою програми EZ-KIT Host завантажити модуль, що виконується, у пам'ять програм сигнального процесора і виконати його. Використовувати кнопку RESET для виконання апаратного скидання стану;

- за допомогою програми Visual DSP Debugger виконати покрокове виконання тієї ж самої програми. Скласти таблицю зміни значень у регістрах процесора;

- зміст командного файлу і таблиці зміни значень у регістрах процесора представити в звіті.

3.2.5 Розробка архітектурного файлу системи:

- одержати завдання у викладача;
- створити на диску свій робочий каталог (наприклад, C:\Work\lab3) ;

- скопіювати в нього файл bld21.exe;
- за допомогою текстового редактора скласти текст програм відповідно до завдання;

- скласти .BAT-файл пакетного виконання завдання;
- у пакетному режимі виконати завдання;
- скласти таблицю розподілення ресурсів системи;

- зміст командного файлу і таблиці представити у звіті.

3.2.6 Підготовка файлу для прошивання ППЗУ:

- одержати завдання у викладача;
- створити на диску свій робочий каталог (наприклад, C:\Work\lab3);
- скопіювати в нього файл spl21.exe;
- за допомогою текстового редактора скласти текст програм відповідно до завдання;
- скласти .BAT-файл пакетного виконання завдання;
- у пакетному режимі виконати завдання;
- одержати 16-річний дамپ розщепленої версії модуля програми, що виконується;
- зміст командного файлу, вихідний текст програми і дамп її версії для запису в ППЗУ представити в звіті.

3.3 Контрольні питання

3.3.1 Склад стенду ADDS-2181 EZ-KIT Lite. Призначення його блоків.

3.3.2 Технічні характеристики стенда ADDS-2181 EZ-KIT Lite.

3.3.3 Призначення і використання програми EZ-KIT Lite Host.

3.3.4 Призначення і використання програми EZ-KIT Lite Monitor.

3.3.5 Доступні способи налагодження ПО для ЦПОС ADSP-2181.

3.3.6 Призначення і використання програми Visual DSP.

3.3.7 Призначення і використання програми BLD21.

3.3.8 Призначення файлу опису архітектури (.SYS).

3.3.9 Особливості використання мови Сі при написанні програм для сигнальних процесорів.

3.3.10 Призначення і використання програми G21.

3.3.11 Склад і використання Run-time Library для мови Сі.

3.3.12 Структура програми мовою Сі для ЦПОС ADSP-2181.

4 ЛАБОРАТОРНА РОБОТА № 4

МОВА АСЕМБЛЕРА ЦПОС ADSP-2181

Мета роботи: вивчити інструкції асемблера процесора ADSP-2181. Придбати практичні навички написання програм для нього і виконання їх на симуляторі Visual DSP Debugger.

4.1 Завдання

4.1.1 Скласти програму по заданій блок-схемі алгоритму (рис. 11.1, а).

4.1.2 Скласти програму по заданій блок-схемі алгоритму (рис. 11.1, б).

4.1.3 Скласти програму по заданій блок-схемі алгоритму (рис. 11.1, в).

4.1.4, 4.1.5, 4.1.6 Складання програми по заданій таблиці зміни значень у регістрах процесора.

4.2 Порядок виконання роботи

4.2.1 Складання програми по заданій блок-схемі алгоритму:

- вивчити задану блок-схему алгоритму;
- у каталозі \Sources знаходиться файл sample.dsp, що містить приклад програми, що працює по заданому алгоритму;
- створити на диску свій робочий каталог, наприклад, C:\work\lab4. Скопіювати в нього файли sample.dsp, asm21.exe, ld21.exe і файл опису архітектури;
- у текстовому редакторі змінити зміст програми sample.dsp у відповідності зі своїм бажанням (творча робота);
- виконати компіляцію і компоновання нової версії файлу sample.exe;
- запустити відладчик Visual DSP і завантажити в інтегроване середовище свою версію файлу, що виконується, sample.exe;
- зробити покрокове виконання програми із середовища симулятора ЦПОС ADSP-2181. Одночасно скласти і заповнити таблицю зміни значень регістрів процесора;
- скласти блок-схему алгоритму програми, заповнивши блоки, що пустують, новим змістом (інструкціями асемблера);

- у результаті лабораторної роботи повинні бути отримані всі дані, необхідні для опису роботи програми sample. Записати загальну кількість кроків (машинних тактів) виконання програми.

4.2.2 Складання програми по заданій таблиці зміни значень у регістрах процесора:

- одержати завдання у викладача. Вивчити задану таблицю зміни значень регістрів процесора;

- скласти блок-схему алгоритму програми, заповнивши блоки необхідним змістом (інструкціями асемблера);

- створити на диску свій робочий каталог, наприклад, C:\work\lab4. Скопіювати в нього файли asm21.exe, ld21.exe і файл опису архітектури;

- у текстовому редакторі створити новий документ, що містить текст програми мовою асемблера ЦПОС ADSP-2181. Зберегти його у вигляді текстового файлу sample.dsp;

- виконати компіляцію і компонування файлу sample.exe;

- запустити відладчик Visual DSP і завантажити в інтегроване середовище свою версію файлу, що виконується, sample.exe;

- зробити покрокове виконання програми із середовища симулятора ЦПОС ADSP-2181. При необхідності зробити потрібні зміни у вихідному тексті і заново виконати компіляцію і компонування програми sample.exe;

- у результаті лабораторної роботи треба отримати версію коректно працюючої програми і всі дані, необхідні для опису ходу її роботи. Записати загальну кількість кроків (машинних тактів) виконання програми.

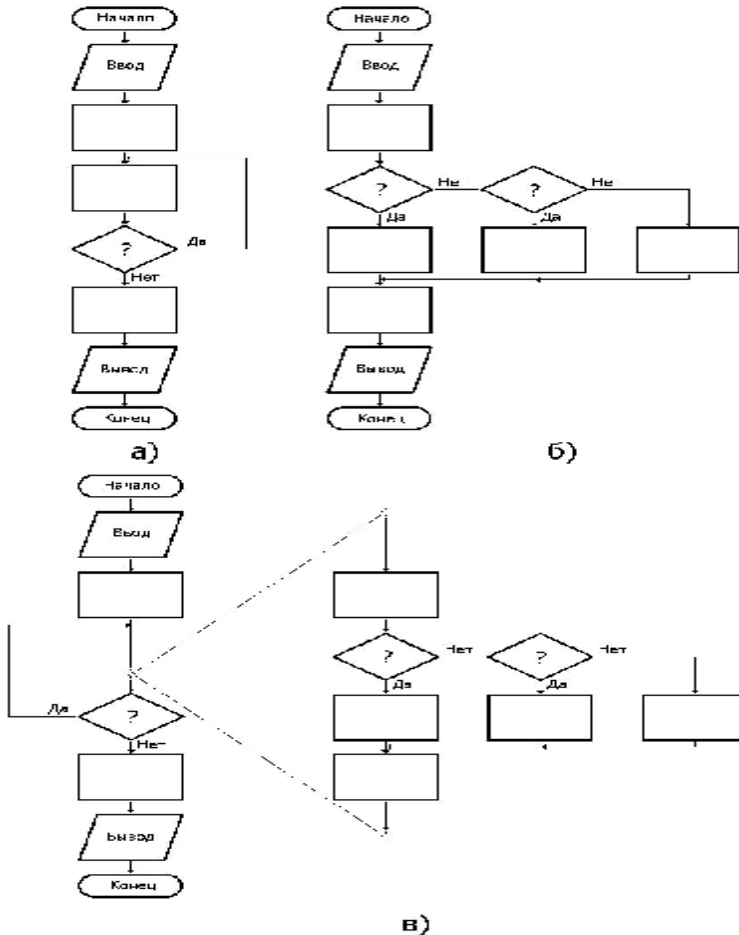


Рисунок 4.1 – Варианты завдань

:

4.3 Контрольні питання

4.3.1 Призначення регістрів процесора ADSP-2181.

4.3.2 Прямая, лінійна непряма і циклічна непряма адресація в ЦПОС сімейства ADSP-21XX.

4.3.3 Пам'ять програм і пам'ять даних процесора ADSP-2181.

- 4.3.4 Переривання ЦПОС ADSP-2181. Регістри керування перериваннями.
- 4.3.5 Організація циклів у ЦПОС сімейства ADSP-21XX.
- 4.3.6 Призначення регістрів ASTAT, SSTAT і MSTAT.
- 4.3.7 Виклик підпрограм і обробка переривань у процесорах сімейства ADSP-21XX.
- 4.3.8 Регістри таймера-лічильника.
- 4.3.9 Послідовні синхронні порти. Керування.
- 4.3.10 Інструкції мови асемблера для блоків ALU, MAC і Shifter.
- 4.3.11 Багатофункціональні інструкції.
- 4.3.12 Інструкції керування ходом програми і пересилання даних асемблера ASM21.

5 ЛАБОРАТОРНА РОБОТА № 5 МАТЕМАТИЧНІ ОБЧИСЛЕННЯ НА ЦПОС ADSP-2181

Мета роботи: вивчити теоретичні зведення, що відносяться до теми лабораторної роботи. Придбати практичні навички проектування програм середньої складності для ЦПОС ADSP-2181.

5.1 Завдання

- 5.1.1 Програма апроксимації функції $y = \sin(x)$.
- 5.1.2 Програма апроксимації функції $y = \sqrt{x}$.
- 5.1.3 Програма апроксимації функції $y = \arctan(x)$.
- 5.1.4 Програма апроксимації функції $y = \log_{10}(x)$.
- 5.1.5 Програма апроксимації функції $y = \ln(x)$.
- 5.1.6 Програма компадування по А-закону (G.911).

5.2 Порядок виконання роботи

- 5.2.1 Програма апроксимації функції $y = \sin(x)$:
 - повторити необхідні для роботи факти з курсу вищої математики;

- одержати завдання у викладача. За допомогою калькулятора розрахувати очікуваний результат;
- створити на диску свій робочий каталог (наприклад, C:\Work\lab5). Скопіювати в нього файли asm21.exe, ld21.exe і файл опису архітектури;
- у каталозі \Sources знайти заготовку програми (файл sin.dsp). Скопіювати її у свій робочий каталог;
- за допомогою текстового редактора змінити текст програми відповідно до завдання;
- за допомогою програм ASM21 і LD21 одержати модуль, що виконується, (sin.exe);
- запустити програму Visual DSP Debugger. Налаштувати режим симуляції процесора ADSP-2181;
- завантажити в інтегроване середовище програму sin.exe і виконати її. Порівняти отриманий результат з очікуваним;
- використовуючи подібний підхід, скласти таблицю значень функції $\sin(x)$ із кроком 15 градусів.

5.2.2 Програма апроксимації функції $y = \sqrt{x}$:

- повторити необхідні для роботи факти з курсу вищої математики;
- одержати завдання у викладача. За допомогою калькулятора розрахувати очікуваний результат;
- створити на диску свій робочий каталог (наприклад, C:\Work\lab5). Скопіювати в нього файли asm21.exe, ld21.exe і файл опису архітектури;
- у каталозі \Sources знайти заготовку програми (файл sqrt.dsp). Скопіювати її у свій робочий каталог;
- за допомогою текстового редактора змінити текст програми відповідно до завдання;
- за допомогою програм ASM21 і LD21 одержати модуль, що виконується, (sqrt.exe);
- запустити програму Visual DSP Debugger. Налаштувати режим симуляції процесора ADSP-2181;
- Завантажити в інтегроване середовище програму sqrt.exe і виконати її. Порівняти отриманий результат з очікуваним;

- використовуючи подібний підхід, скласти таблицю значень функції квадратного кореня для цілих чисел на відрізьку [1; 100] з довільним кроком (не менш 20 значень).

5.2.3 Програма апроксимації функції $y = \arctan(x)$:

- повторити необхідні для роботи факти з курсу вищої математики;
- одержати завдання у викладача. За допомогою калькулятора розрахувати очікуваний результат;
- створити на диску свій робочий каталог (наприклад, C:\Work\lab5). Скопіювати в нього файли asm21.exe, ld21.exe і файл опису архітектури;
- у каталозі \Sources знайти заготівку програми (файл arctan.dsp). Скопіювати її у свій робочий каталог;
- за допомогою текстового редактора змінити текст програми відповідно до завдання;
- за допомогою програм ASM21 і LD21 одержати модуль, що виконується, (arctan.exe);
- запустити програму Visual DSP Debugger. Налаштувати режим симуляції процесора ADSP-2181;
- завантажити в інтегроване середовище програму arctan.exe і виконати її. Порівняти отриманий результат з очікуваним;
- використовуючи подібний підхід, скласти таблицю значень функції арктангенса для цілих чисел на відрізьку [-50; 50] з довільним кроком (не менш 20 значень).

5.2.4 Програма апроксимації функції $y = \log_{10}(x)$:

- повторити необхідні для роботи факти з курсу вищої математики;
- одержати завдання у викладача. За допомогою калькулятора розрахувати очікуваний результат;
- створити на диску свій робочий каталог (наприклад, C:\Work\lab5). Скопіювати в нього файли asm21.exe, ld21.exe і файл опису архітектури;
- у каталозі \Sources знайти заготівку програми (файл log.dsp). Скопіювати її у свій робочий каталог;
- за допомогою текстового редактора змінити текст програми відповідно до завдання;

- за допомогою програм ASM21 і LD21 одержати модуль, що виконується, (log.exe);
- запустити програму Visual DSP Debugger. Налаштувати режим симуляції процесора ADSP-2181;
- завантажити в інтегроване середовище програму log.exe і виконати її. Зверити отриманий результат з очікуваним;
- використовуючи подібний підхід, скласти таблицю значень функції десяткового логарифма для цілих чисел на відрізку [1; 100] з довільним кроком (не менш 20 значень).

5.2.5 Програма апроксимації функції $y = \ln(x)$.

- повторити необхідні для роботи факти з курсу вищої математики;
- одержати завдання у викладача. За допомогою калькулятора розрахувати очікуваний результат;
- створити на диску свій робочий каталог (наприклад, C:\Work\lab5). Скопіювати в нього файли asm21.exe, ld21.exe і файл опису архітектури;
- у каталозі \Sources знайти заготовку програми (файл log.dsp). Скопіювати її у свій робочий каталог;
- за допомогою текстового редактора змінити текст програми відповідно до завдання;
- за допомогою програм ASM21 і LD21 одержати модуль, що виконується, (log.exe);
- запустити програму Visual DSP Debugger. Налаштувати режим симуляції процесора ADSP-2181;
- завантажити в інтегроване середовище програму log.exe і виконати її. Порівняти отриманий результат з очікуваним;
- використовуючи подібний підхід, скласти таблицю значень функції натурального логарифма для цілих чисел на відрізку [1; 100] з довільним кроком (не менш 20 значень).

5.2.6 Програма компонування по А-закону (G.911):

- вивчити необхідні для роботи факти, що містяться в стандарті G.911 організації ITU-T (колишня CCITT);
- одержати завдання у викладача. За допомогою калькулятора розрахувати очікуваний результат;

- створити на диску свій робочий каталог (наприклад, C:\Work\lab5). Скопіювати в нього файли asm21.exe, ld21.exe і файл опису архітектури;
- у каталозі \Sources знайти заготовки програм компонування і експондування (файли a_compre.dsp і a_expand.dsp). Скопіювати їх у свій робочий каталог;
- за допомогою текстового редактора змінити вихідний текст програми a_compre.dsp відповідно до завдання;
- за допомогою програм ASM21 і LD21 одержати модуль, що виконується, (a_compre.exe);
- запустити програму Visual DSP Debugger. Налаштувати режим симуляції процесора ADSP-218;
- завантажити в інтегроване середовище програму a_compre.exe і виконати її. Порівняти отриманий результат з очікуваним;
- за допомогою текстового редактора змінити вихідний текст програми a_expand.dsp відповідно до завдання;
- за допомогою програм ASM21 і LD21 одержати модуль, що виконується, (a_expand.exe);
- запустити програму Visual DSP Debugger. Налаштувати режим симуляції процесора ADSP-2181;
- завантажити в інтегроване середовище програму a_expand.exe і виконати її. Порівняти отриманий результат з очікуваним;
- зробити висновки про доцільність використання такого способу збільшення динамічного діапазону представлення рівня сигналу.

5.3 Контрольні питання

- 5.3.1 Числові ряди. Апроксимація функції синуса.
- 5.3.2 Числові ряди. Апроксимація функції квадратного кореня.
- 5.3.3 Числові ряди. Апроксимація функції арктангенса.
- 5.3.4 Числові ряди. Апроксимація функції логарифма.
- 5.3.5 Векторна алгебра. Додавання матриць.
- 5.3.6 Векторна алгебра. Множення матриць.
- 5.3.7 Лінійна імпульсно-кодова модуляція.

5.3.8 Стандарт G.911. Компадування по А-закону.

5.3.9 Стандарт G.911. Компадування по μ -закону.

5.3.10 Масштабування. Вибір формату представлення ІКМ-даних.

5.3.11 Використання файлів для збереження даних.

5.3.12 Диференціальна імпульсно-кодова модуляція.

ЛАБОРАТОРНА РОБОТА № 6 ПРОЕКТУВАННЯ FIR-ФІЛЬТРІВ НА ЦПОС ADSP-2181

Мета роботи: ввчити методику проектування цифрових фільтрів з кінцевою імпульсною характеристикою (FIR-фільтрів). Придбати практичні навички цифрової фільтрації. Ознайомитися з принципами ЦОС, що використовуються при проектуванні цифрових фільтрів.

6.1 Завдання

6.1.1 Низькочастотний нерекурсивний фільтр. Проектування методом вікна.

6.1.2 Високочастотний нерекурсивний фільтр. Проектування методом вікна.

6.1.3 Смуговий нерекурсивний фільтр. Проектування методом вікна.

6.1.4 Низькочастотний нерекурсивний фільтр. Проектування методом Ремеза.

6.1.5 Високочастотний нерекурсивний фільтр. Проектування методом Ремеза.

6.1.6 Смуговий нерекурсивний фільтр. Проектування методом Ремеза.

6.2 Порядок виконання роботи

6.2.1 Проектування нерекурсивного фільтра методом вікна:

- запустити програму ADSP-218x Navigator (файл 218xNavigator.exe);
- відкрити вікно [125];
- запустити приклад 22. Ознайомитися з принципом роботи FIR-фільтра;

- запустити приклад 23. Ознайомитися зі змістом програми цифрової фільтрації для ЦПОС ADSP-2181;
- одержати завдання на проектування у викладача;
- створити на диску свій робочий каталог, наприклад, C:\work\lab6. Скопіювати в нього файли todat.com, FIR.m, firdemo.dsp, asm21.exe, ld21.exe, файл опису архітектури і файли дані програми FIRDEMO;
- за допомогою програми MatLAB розрахувати порядок і значення коефіцієнтів фільтра методом вікна;
- за допомогою програми todat.com перетворити десяткові значення коефіцієнтів фільтра у формат 16.0. У результаті на диску повинен з'явитися файл data.dat, що містить коректні значення коефіцієнтів нерекурсивного фільтра;
- у текстовому редакторі змінити зміст програми firdemo.dsp відповідно до отриманих результатів;
- виконати компіляцію і компонування нової версії файлу FIRDEMO, що використовує як файл даних отриманий файл коефіцієнтів фільтра;
- скопіювати в робочий каталог програми EZ-KIT Lite Host свій модуль, що виконується, firdemo.exe і файл data.dat. Виключити комп'ютер;
- зібрати схему експерименту (рис. 13.1). Ввімкнути комп'ютер. При необхідності виконати налаштування звукової карти таким чином, щоб вона забезпечувала одночасний ввід/вивід аудіоданих (режим full-duplex);
- запустити програму SpectraLAB. Налаштувати режим Real Time (Mode -> Real Time);
- Запустити програму EZ-KIT Lite Host, де активізувати демонстраційну програму цифрової фільтрації (Filter). Вибираємо номер фільтра, що відповідає розробленому;
- запустити вбудований генератор звуку програми SpectraLAB (Utilities -> Generator). Налаштувати бажаний режим роботи генератора і максимальний рівень сигналу на лінійному виході звукової карти;
- переключачи через <Alt> + <Tab> можна в реальному масштабі часу змінювати режими роботи запущених програм. Мета – одержати на екрані програмного спектрографа чітку картину зміни спектра сигналу на вході і виході стенда EZ-KIT Lite (плюс АЧХ звукової карти). Це - результат цифрової фільтрації;

- за допомогою натискання <PrintScreen> зробити "знімки" зображення, які можна використовувати для ілюстрації звіту. Крім них, у звіт додати список значень коефіцієнтів FIR-фільтра.

6.2.2 Проектування нерекурсивного фільтра методом Ремеза:

- запустити програму ADSP-218x Navigator (файл 218xNavigator.exe);

- відкрити вікно [125];

- запустити приклад 22. Ознайомитися з принципом роботи FIR-фільтра;

- запустити приклад 23. Ознайомитися зі змістом програми цифрової фільтрації для ЦПОС ADSP-218;

- одержати завдання на проектування у викладача;

- створити на диску свій робочий каталог, наприклад, C:\work\lab6. Скопіювати в нього файли todат.com, FIR.m, firdemo.dsp, asm21.exe, ld21.exe, файл опису архітектури і файли дані програми FIRDEMO;

- за допомогою програми MatLAB розрахувати порядок і значення коефіцієнтів фільтра методом Ремеза;

- за допомогою програми todат.com перетворити десяткові значення коефіцієнтів фільтра у формат 16.0. В результаті на диску повинен з'явитися файл data.dat, що містить нові значення коефіцієнтів нерекурсивного фільтра;

- у текстовому редакторі змінити зміст програми firdemo.dsp відповідно до отриманих результатів;

- виконати компіляцію і компонування нової версії файлу FIRDEMO, що використовує як файл даних отриманий новий файл коефіцієнтів фільтра;

- скопіювати в робочий каталог програми EZ-KIT Lite Host модуль, що виконується, firdemo.exe і файл data.dat. Виключити комп'ютер;

- зібрати схему експерименту (рис. 6.1). Ввімкнути комп'ютер. При необхідності виконати налаштування звукової карти таким чином, щоб вона забезпечувала одночасний ввід/вивід аудіоданих (режим full-duplex);

- запустити програму SpectraLAB. Налаштувати режим Real Time (Mode -> Real Time);

- запустити програму EZ-KIT Lite Host, де активізувати демонстраційну програму цифрової фільтрації (Filter). Вибірємо номер фільтра, що відповідає розробленому;

- запустити вбудований генератор звуку програми SpectraLAB (Utilities -> Generator). Налаштувати бажаний режим роботи

генератора і максимальний рівень сигналу на лінійному виході звукової карти;

- переключачи через <Alt> + <Tab> можна в реальному масштабі часу змінювати режими роботи запущених програм. Мета – одержати на екрані програмного спектрографа чітку картину зміни спектра сигналу на вході і виході стенда EZ-KIT Lite (плюс АЧХ звукової карти). Це –результат цифрової фільтрації;

- за допомогою натискання <PrintScreen> зробити "знімки" зображення на екрані, які можна використовувати для ілюстрації звіту. Крім них, у звіт додати список значень коефіцієнтів FIR-фільтра.

6.3 Контрольні питання

6.3.1 Класифікація пристроїв фільтрації. Області використання аналогових і цифрових фільтрів.

6.3.2 Згортка сигналу функції відгуку у часовій області.

6.3.3 Ряд Фур'є. Дискретний ряд Фур'є.

6.3.4 Перетворення Фур'є. Дискретне перетворення Фур'є.

6.3.5 Принцип дії і властивості FIR-фільтра.

6.3.6 Теорема Найквіста-Котельникова.

6.3.7 Принцип дії і властивості FIR -фільтрів.

6.3.8 Основні характеристики FIR-фільтра.

6.3.9 Проектування FIR-фільтра. Метод вікна.

6.3.10 Проектування FIR-фільтра. Метод Ремеза.

6.3.11 Блок-схема пристрою цифрової фільтрації.

6.3.12 Організація програми FIRDEMO.

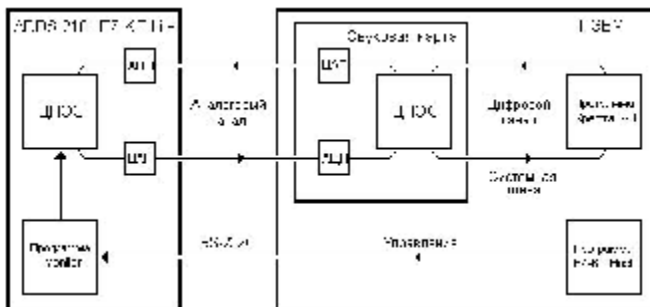


Рисунок 6.1 - Схема лабораторного стенда

ЛІТЕРАТУРА

1. Бондарев В.Н., Трестер Г., Чернега В.С. Цифровая обработка сигналов: методы и средства: Учеб. пособие для вузов.- Севастополь: Изд-во СевГТУ, 1999.
2. Руководство пользователя по сигнальным микропроцессорам семейства ADSP-21XX/ Пер. с англ. О.В. Луневой; Под ред. А.Д. Викторова; Санкт-Петербургский государственный электротехнический университет.- Санкт-Петербург, 1997.
3. Шрюфер Э. Обработка сигналов: цифровая обработка дискретизированных сигналов: Учебник/ Под ред. В.П. Бабака.- К.: Либідь, 1995.
4. Гук М. Интерфейсы ПК: справочник – СПб: Питер Ком, 1999. – 416 с.
5. ADSP-21XX Family EZ-KIT Lite Reference Manual, Analog Devices, 1995.
6. 21xx Family Tools Release Notes (for the ADSP-21xx DSP), Analog Devices, 1998
7. Visual DSP Debugger Tutorial (for the ADSP-21xx Family DSP), Analog Devices, 1998.
8. Visual DSP Debugger Guide and Reference, Analog Devices, 1998.
9. ADSP-2181 EZ-KIT Lite Evaluation System Manual, Analog Devices, 2001.
10. ADSP-21XX Family User's Manual, Prentice Hall, 1994.
11. Бойко В.И. и др. Схемотехника электронных систем. Микропроцессоры и микроконтроллеры.- СПб.: БХВ-Петербург 2004.- 464 с.
12. Digital Signal Processing Applications Using the ADSP-21XX Family, Vol.1, Prentice Hall, 1992.
13. Digital Signal Processing Applications Using the ADSP-21XX Family, Vol.2, Prentice Hall, 1994.
14. Analog Devices' Data Sheets: AD1847 (Rev.B, 1996), ADSP-21XX Family Development Tools (Rev.A, 1995.)
15. FIR-filters without tears // Analog Dialogue #17-2, 1983.

Додаток А Мікропроцесор ADSP-2181

А.1 Внутрішня будова мікропроцесора ADSP-2181

Мікропроцесор **ADSP-2181** є базовою моделлю серії. ADSP-2181 має 81920 (80K) байт вбудованої пам'яті статичного типу (16384 (16K) 24-розрядних слів ПП і 16384 (16K) 16-розрядних слів ПД). Структурно-функціональна організація мікропроцесора ADSP-2181 зображена на рис.А.1.

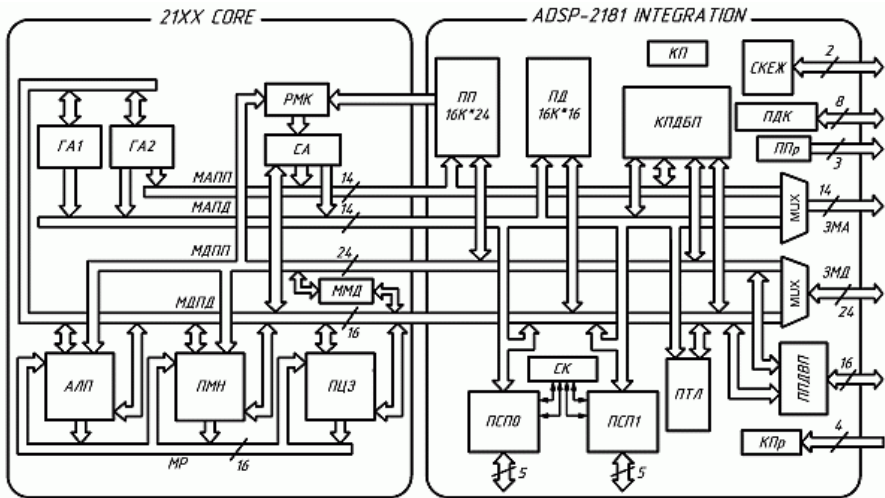


Рисунок А.1 – Структурно-функціональна організація ЦПІОС ADSP-2181

Мікропроцесор ADSP-2181 об'єднує на кристалі дві підсистеми: загальне для всіх моделей серії обчислювальне ядро 21XX Core і архітектурні розширення (ADSP-2181 Integration). Склад і характеристики архітектурних розширень залежать від моделі мікропроцесора. В таблиці А.1 перелічені основні структурні елементи мікропроцесора ADSP-2181.

Обчислювальне ядро мікропроцесора містить три операційних блока: АЛП, ПМН і ПЦЗ. За допомогою магістралі результату (МР) проміжні результати з виходу одного блоку можуть бути передані на вхід іншого. Паралельна робота операційних блоків не підтримується.

Наявність незалежних генераторів адреси даних (ГА1, ГА2) і СА дозволяє значно скоротити час виконання команд переміщення даних і керування програмою (цикли, виклики підпрограм, умовні переходи та ін.). Глибина вкладеності апаратно реалізованих циклів не повинна перевищувати чотирьох. Автомат не може обслуговувати рекурсивні виклики підпрограм, число яких перевищує розмір стеку.

Таблиця А.1 – Архітектурні елементи ADSP-2181

Назва (укр.)	Назва (англ.)	Призначення
ГА1, ГА2	DAG1, DAG2	Генератори адреси даних
АЛП	ALU	Арифметично-логічний пристрій
ПМН	MAC	Пристрій множення / накопичування
ПЦЗ	Shifter	Пристрій циклічного зсуву
МАПП	PMA bus	Магістраль адреси пам'яті програм (14 біт)
МАПД	DMA bus	Магістраль адреси пам'яті даних (14 біт)
МДПП	PMD bus	Магістраль даних пам'яті програм (24 біт)
МДПД	DMD bus	Магістраль даних пам'яті даних (16 біт)
МР	R bus	Магістраль результату (16 біт)
ММД	Bus exchange	Міст магістралей даних
РМК	Instruction register	Регістр команд
СА	Program sequencer	Системний автомат
ПП	PM	Пам'ять програм (48Кбайт)
ПД	DM	Пам'ять даних (32Кбайт)
ПСПО, ПСП1	SPORT0, SPORT1	Програмовані синхронні порти
СК	Companding circuit	Схема компандування (ITU-T: G.711)
КПДБП	BDMA	Контролер ПДП байтової пам'яті
ПТЛ	Timer	Програмований таймер-лічильник
СКЕЖ	Power down circuit	Схема керування електроживленням
ПДК	Programmable I/O	Програмовані дискретні канали (8 біт)
ППр	Programmable flags	Програмовані прапори
ЗМА	Address bus	Зовнішня магістраль адреси (14 біт)
ЗМД	Data bus	Зовнішня магістраль даних (24 біт)
ППДВП	IDMA	Порт ПДП внутрішньої пам'яті (16 біт)
КІр	Interrupt controller	Контролер переривань

А.2 Арифметико - логічний пристрій

Арифметично-логічний пристрій призначений для виконання основних адитивних, логічних, бітових операцій і примітиву ділення. Набір регістрів АЛП складається з двох копій регістрів AX0, AX1, AY0, AY1, AR, AF (загалом – дванадцять 16-розрядних регістрів). В табл.А.2 перелічені операційні регістри АЛП.

Таблиця А.2 – Регістри АЛП

Регістр / адреса	RGP	DREG	Біт	Призначення
AX0	00	0000	16	Молодше слово операнда X
AX1	00	0001	16	Старше слово операнда X
AY0	00	0100	16	Молодше слово операнда Y
AY1	00	0101	16	Старше слово операнда Y
AR	00	1010	16	Результат
AF	-	-	16	Проміжний результат

Структурно-функціональна організація АЛП приведена на рис.А.2.

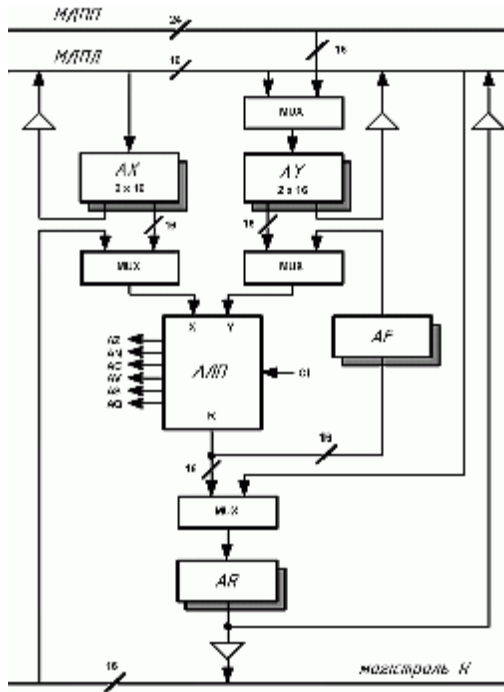


Рисунок А.2 – Арифметично-логічний пристрій ADSP-2181

А.3 Пристій множення / накопичування

Пристрій множення / накопичення призначений для виконання операцій множення і множення з накопиченням. Набір регістрів ПМН складається з двох копій регістрів MX0, MX1, MY0, MY1, MR0, MR1,

MF (16 біт) і MR2 (8 біт). Загалом – 16 регістрів. Ввімкнення/вимикання тіньового набору регістрів виконується командами ENA SEC_REG і DIS SEC_REG відповідно.

Додатковий 8-розрядний регістр MR2 використовується для збереження результату виконання операції множення з накопиченням від втрат, пов'язаних з переповненням. Використання регістру MR2 захищає результат навіть тоді, коли виникло $2^8 = 256$ переповнень розрядної сітки.

В табл. А.3 перелічені регістри даних ПМН.

Таблиця А.3 – Регістри ПМН

Регістр / адреса	RGP	DREG	Біт	Призначення
MX0	00	0010	16	Молодше слово операнда X
MX1	00	0011	16	Старше слово операнда X
MY0	00	0110	16	Молодше слово операнда Y
MY1	00	0111	16	Старше слово операнда Y
MR0	00	1011	16	Перше слово результату
MR1	00	1101	16	Друге слово результату
MR2	00	1110	8	Третє слово результату
MF	-	-	16	Проміжний результат

Структурно-функціональна організація ПМН приведена на рис.А.3.

В ПМН використовується матричний спосіб виконання операції множення, який дозволяє отримати результат впродовж одного МЦ. Після множення 16-розрядних операндів отримують 32-розрядний результат. В залежності від формату даних, що перемножувались, результат множення Р може бути записаний в регістр результату MR двома способами:

– якщо виконувалось множення цілих чисел (формат 16.0), то Р просто копіюється в MR;

– якщо перемножувались дробові числа (формат 1.15), то в MR заноситься результат множення, зсунутий на один розряд вліво (див.рис.А.4).

Спосіб розміщення результату ПМН залежить від стану прапору M_MODE (четвертий біт регістру MSTAT).

А.5 Пристрій циклічного зсуву

Пристрій циклічного зсуву призначений для виконання операцій зсуву, нормалізації та знаходження порядку чисел у форматі 1.15.

SEC_REG і DIS SEC_REG відповідно. В табл.А.4 перелічені регістри даних ПЦЗ.

Таблиця А.4 – Регістри ПЦЗ

Регістр / адреса	RGP	DREG	Біт	Призначення
SI	00	1000	16	Вхідний операнд
SE	00	1001	8	Порядок зсуву
SB	11	0110	5	Зсув блоку операндів
SR0	00	1110	16	Перше слово результату ПЦЗ
SR1	00	1111	16	Друге слово результату ПЦЗ

Структурно-функціональна організація ПЦЗ приведена на рис.А.6. Операційний пристрій зсуву складається з 16 регістрів розрядністю 32 біти. До цього регістрового файлу вводиться 16-розрядне значення, положення якого завдають керуючий 8-розрядний код C і сигнали HI / LO. Данні в ПЦЗ надходять через регістр SI.

Регістр результату SR має розрядність 32 біти, він складається двох 16-розрядних регістрів SR0, SR1. Він має зворотній зв'язок з логічним пристроєм OR/PASS для забезпечення 32-розрядних операцій зсуву. Якщо в команді вказана операція SR OR, нове значення результату записується після логічного АБО з попереднім змістом регістру SR.

Регістр SE (8 біт) містить значення порядку зсуву. Негативне значення порядку призводить до зсуву даних вліво, позитивне значення – до зсуву вправо. Значення цього регістру може бути визначено після виконання операції визначення порядку EXP (SFT25 – SFT30) або задано безпосередньо.

Регістр SB (5 біт) призначений для знаходження максимального порядку блоку чисел з плаваючою крапкою (операції SFT31, SFT32).

Після зсуву всі біти праворуч від вхідного значення в регістрі результату заповнюються нулями, а біти ліворуч – додатковим бітом X. Значення біту X залежить від команди, що виконується. Це може бути самий старший біт значення регістру SI (ASHIFT), біт AC регістру ASTAT (зсув значення AR, в якому зберігається результат операції, яка призвела до переповнення (модифікатор (HIX))) або нуль (LSHIFT).

Модифікатори (HI) і (LO) вказують місце положення вхідного 16-розрядного значення в 32-розрядному регістровому файлі ПЦЗ. Наприклад, якщо треба виконати зсув 32-розрядного слова вправо.

– для старших 16 біт вхідного значення виконується операція ASHIFT з модифікатором (HI). Запис результату – без операції SR OR;

—для молодших 16 біт вхідного слова виконується операція LSHIFT з модифікатором (LO). Запис результату – з операцією SR OR.

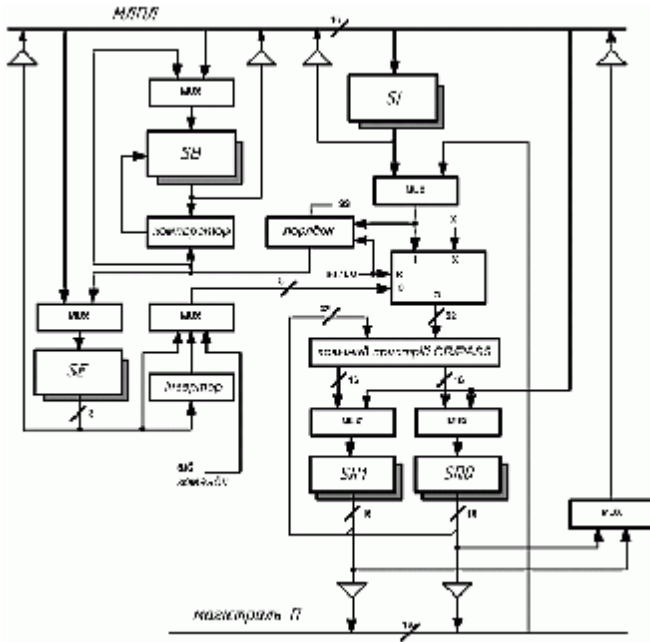


Рисунок А.6 – Пристрій зсуву ADSP-2181

А.6 Генератори адреси

Майже усі ЦПОС, у тому числі ADSP-2181, широко використовують непряму адресацію тому, що для вирішення переважної більшості задач ЦОС часто необхідно використовувати буфери – масиви однорідних даних. Використання непрямой адресації в ADSP-2181 забезпечують генератори адреси ГА1, ГА2. За їх допомоги можливо швидко отримати доступ до елементів масиву даних.

Генератор адреси ГА1 генерує тільки адреси даних, що зберігаються у ПД, але забезпечує додаткову функцію – розміщення бітів слова адреси у зворотному порядку (біт-інверсна адресація). Генератор ГА2 може формувати адресу даних для обох просторів пам'яті – ПД і ПП.

Структурно-функціональна організація ГА приведена на рис.А.7.

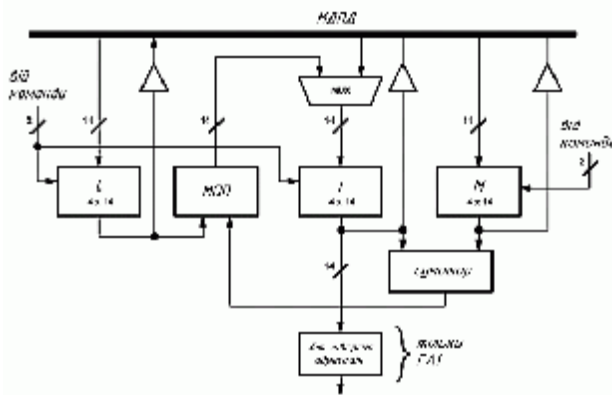


Рисунок А.7 – Генератор адреси ADSP-2181

А.7 Системный автомат

Системний автомат керує процесом виконання команд програми. Він відповідає за апаратну підтримку основних режимів роботи мікропроцесора:

- послідовне виконання команд;
- виконання циклів DO UNTIL (у тому числі вкладених);
- реалізацію умовних і безумовних переходів (JUMP);
- обробку викликів переривань (у тому числі вкладених);
- обслуговування викликів підпрограм CALL;
- повернення керування після виконання підпрограми RTS;
- повернення керування з процедури обслуговування переривання RTI;
- керування виконанням багатофункціональних команд;
- апаратне опитування прапора FI;
- включення / виключення режиму економічного електроживлення IDLE;
- включення / виключення режиму зменшення частоти МЦ (IDLE n);
- включення / виключення режиму GO, таймеру та ін.

Структурно-функціональна організація системного автомату приведена на рис.А.8. На вхід мікропрограмного автомату (МПА) СА надходять логічні умови (ЛУ), які вказують місцезнаходження адреси

наступної команди програми. Ця адреса може зберігатися в одному з п'яти джерел:

- пристрій інкременту лічильника команд (PC);
- стек лічильника команд (стек PC);
- реєстр команд ;
- контролер переривань ;
- індексний реєстр ГА1.

Лічильник команд PC має розрядність 14 біт і містить адресу поточної команди. На лінійних ділянках програми значення PC нарощується пристроєм інкременту. Лічильник команд зв'язаний зі стеком PC, в якому може зберігатися до шістнадцяти 14-розрядних слів. Під час виконання викликів підпрограм, організації циклів DO UNTIL або обслуговування переривання, до стеку лічильника команд автоматично записується вихідне значення пристрою інкрементора (адреса наступної команди програми). По закінченні роботи цикл, підпрограма та процедура обслуговування переривання повертають керування основній програмі і зміст стеку лічильника команд автоматично відновлюється.

Значення, яке зберігається на вершині стеку PC ідентифікує псевдореєстр TOPPCSTACK. За його допомогою воно може бути прочитане / записане програмними засобами.

Лічильник CNTR використовується циклом DO UNTIL і має розрядність 14 біт. Якщо потрібно організувати N повторів виконання циклічної ділянки програми, до реєстру CNTR записують число N-1. Після запису у реєстр CNTR, його значення автоматично завантажується до стеку лічильника циклу.

Стек лічильника призначений для зберігання не більше чотирьох 14-розрядних слів – лічильників вкладених циклів. На вершині циклу міститься лічильник циклу, який має найбільший рівень вкладеності. Для того, щоб змінити значення поточного лічильника циклу, у програмі можна використовувати спеціальний реєстр завантаження лічильника циклу OWRCNTR. Після запису у реєстр OWRCNTR, його значення не завантажується до стеку лічильника циклу, а передається у реєстр CNTR.

Стек циклів складається з чотирьох 18-розрядних реєстрів. Він містить адреси останніх команд циклічних ділянок програми (14 біт) та код логічної умови закінчення тимчасово затриманих циклів (4 біти). Код ЛУ закінчення циклу TERM передається у контексті машинної команди організації циклу DO UNTIL.

Під час передачі / повертання керування основній програмі мікропроцесор відновлює зміст реєстрів стану, зі стеку PC вилучає останню команду.

Додаток Б

Опис станда ADDS-2181 EZ-KIT Lite

Апаратна частина учбового станда являє собою друковану плату розміром 3.5 x 5.5 дюймів. На ній знаходяться ПЦОС ADSP-2181, ППЗУ, звуковий кодек AD1847, різні додаткові ланцюги і рознімання (рис. Б.1). Він може використовуватися як платформа для розробки нових програм для ADSP-2181.

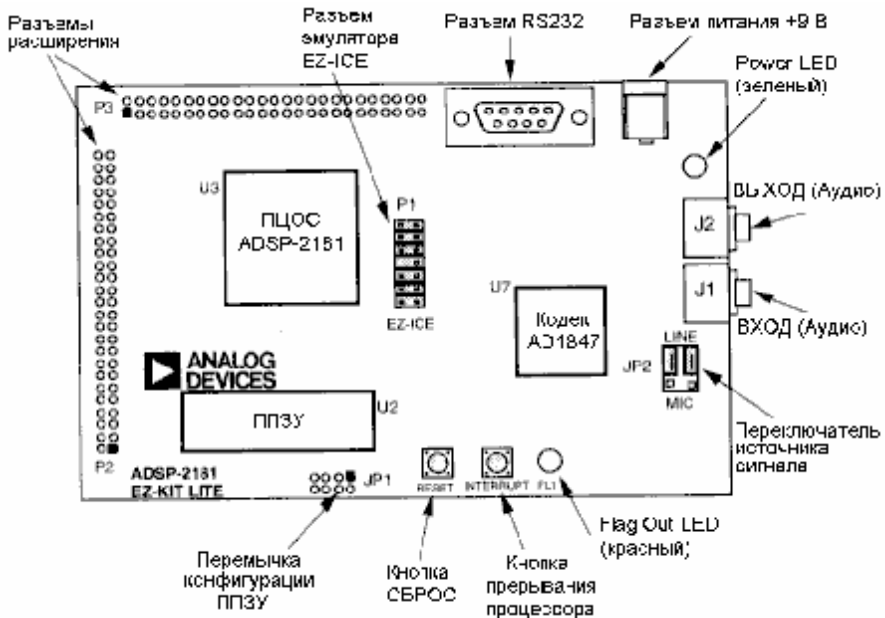


Рисунок Б.1— Зовнішній вигляд плати станда ADDS-2181 EZ-KIT Lite

Стенд EZ-KIT Lite - приклад мінімальної реалізації системи ЦОС на процесорі ADSP-2181. ППЗУ з'єднано з процесором через порт BDMA. Цей інтерфейс використовує тільки вісім з 24 ліній для передачі даних (D8-D15). Вісім інших каналів даних (D16-D23) використовуються для передачі додаткових розрядів адреси. Це дозволяє ADSP-2181 адресувати до 32 Мбіт (4 Мбайт) пам'яті. Гніздо на друкованій платі може бути використане для установки мікросхем ППЗУ ємністю від 256 до 8 Мбіт. Перемичка JP1 забезпечує

необхідний спосіб керування висновками 3 і 30 ці гнізда, у залежності від мікросхеми ППЗУ, яка використовується. Процесор, підключений так, що після скидання системи він автоматично завантажує програму з ППЗУ і починає її виконувати. Порт IDMA процесором не використовується. Усі сигнали IDMA доступні на розніманні P3.

Кодек AD1847 з'єднаний із ПЦОС через SPORT0. Цей високошвидкісний синхронний послідовний порт передає дані, команди керування та інформацію про стан між ADSP-2181 і кодеком. Є Можливість відключення кодека, якщо послідовний порт планується використовувати для іншої мети. Для цього може використовуватися сигнал CODECDIS, доступний на розніманні P3. Коли цей сигнал має низький логічний рівень, кодек заблокований, а сигнали знаходяться в стані високого імпедансу.

Канал SPORT1 використовується для зв'язку з хост-машиною через інтерфейс RS-232 (U5). Висновки Flag In і Flag Out сигналізують про прийом/передачу даних. Прийняті дані також йдуть до IRQ1, так що процесор може знайти їх без опитування стану висновку Flag In. Програма прийому/передачі ПЦОС емулює роботу універсального асинхронного прийомоприймача (УАПП, англ. UART) для забезпечення відповідного протоколу асинхронного послідовного зв'язку зі швидкістю передачі 9600 біт/сек.

Таблиця Б.1 – Технічні характеристики стенда ADDS-2181 EZ-KIT Lite

Основні характеристики	
Процесор	ADSP-2181KS-133 (частота 33 МГц, зовнішнє джерело синхроімпульсів 16,667 МГц)
Аналоговий інтерфейс	Звуковий стерео кодек AD1847
Аналогові входи	Одна пара лінійних стерео входів (рівень до 2 В)
	Одна пара мікрофонних стерео входів (рівень до 20 мВ)
Аналоговий вхід	Одна пара лінійних стерео виходів (рівень до 1 В)
Джерело постійного струму	8-10 В, 300 мА
Умови експлуатації	Температура 0-700°C, відносна вологість 10-90 %

Призначення розніманих	
J1	Вхід звуковий, стерео (3,5 мм).
J2	Вихід звуковий, стерео (3,5 мм).
J3	Цифрове 9-контактне рознімання (інтерфейс RS-232 для зв'язку з ЕОМ).
J4	Штирькове рознімання напруги +9В (зовнішній діаметр 5,5 мм (+ джерело напруги), внутрішній діаметр 2 мм (- джерело напруги).
JP1	Використовується для конфігурування ППЗУ (8 контактів).
JP2	Використовується для налаштування рівня вхідного сигналу, що надходить на рознімання J1 (лінійний вхід або мікрофон). Має 6 контактів, розділених на дві групи (правий і лівий канал). Середній контакт кожної групи приєднаний до входу Line 1 кодека AD1847. Для більш точного регулювання рівнем вхідного сигналу можна використовувати додатковий ланцюг узгодження.
P1	14-контактний роз'єм для підключення внутрішньосхемного емулятора EZ-ICE. Вихід 7 відіграє роль ключа (повинен бути вилучений).
P2, P3	50-контактні рознімання розширення. Можуть використовуватися для розширення функціональних можливостей стенда або для тестування.
U2	Гніздо для встановлення мікросхеми ППЗУ. Можуть бути використані мікросхеми 27C512 (ємність 64 Кбайт), 27C010 (128Кбайт), 27C256 (32Кбайт), 27C020 (256Кбайт), 27C040 (512Кбайт) або 27C080 (1Мбайт). Тип установлюваної мікросхеми ППЗУ задається за допомогою JP1.
R2 8	Місце для резистора з нульовим опором (перемички). Якщо цей резистор встановлений, то процесор одержує можливість самостійно робити скидання апаратної частини стенда під керуванням програми. Для цього в програмі для ADSP-2181 необхідно використовувати можливість конфігурування PF0.
R2 9	Місце для перемички. Якщо її встановити і видалити X3 і C37, то кодек зможе використовувати сигнал ADSP-2181 CLKOUT замість власного кварцового генератора 24.576 МГц. У цьому випадку, для узгодження процесора і кодека, необхідно замінити X1 на більш низьке значення частоти.
Призначення перемикачів	
S1	Кнопка скидання. Натискання цієї кнопки призводить до скидання ADSP-2181 і AD1847. Має захист від деренчання контактів.
S2	Кнопка переривання. Натискання цієї кнопки змушує процесор виконати процедуру переривання IRQE (якщо воно передбачено в програмі і вектор переривання не змінив своє значення). Має захист від деренчання контактів.