**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG ĐIỆN - ĐIỆN TỬ**

**TRUNG TÂM ĐÀO TẠO THỰC HÀNH ĐIỆN – ĐIỆN TỬ**

**KẾT QUẢ THÍ NGHIỆM**

**XỬ LÝ TÍN HIỆU – ET4020**

|  |  |  |
| --- | --- | --- |
|  | **Họ và tên** | Nguyễn Trọng Phúc |
|  | **MSSV** | 20224099 |
|  | **Lớp - khóa** | Điện tử 09 – K67 |
|  | **Mã lớp thí nghiệm** | 753989 |
|  | **Tên lớp thí nghiệm** | Xử lý tín hiệu số |

# Hà Nội, 2024

# MỤC LỤC

[BOARD TN 1: Thí nghiệm Xử lý tín hiệu - Digital Signal Processor. 2](#_Toc164546256)

[1.1. U1: Làm quen với hệ thống thí nghiệm DSP. 2](#_Toc164546257)

[1.1.1. Ex 1: Các kiến thức cơ bản. 2](#_Toc164546258)

[1.1.2. Ex 2: Tìm hiểu tổng quan về bo mạch thí nghiệm DSP. 2](#_Toc164546259)

[Giới thiệu bo mạch Thí nghiệm 2](#_Toc164546260)

[Làm quen với bảng mạch bằng chương trình DSP 2](#_Toc164546261)

[Làm việc với ứng dụng thứ 2 4](#_Toc164546262)

[1.1.3. Ex 3: Tìm hiểu Tool lập trình và cấu trúc Project. 4](#_Toc164546263)

[Trình Phát triển và Cấu trúc Dự án 4](#_Toc164546264)

[1.2. U2: Kiến trúc DSP. 9](#_Toc164546265)

[1.2.1. Ex 1: Bộ xử lý số học. 9](#_Toc164546266)

[Biểu diễn số thập phân có dấu ở dạng Q14 9](#_Toc164546267)

[Chuyển số Nhị phân sang Thập phân 10](#_Toc164546268)

[So sánh phép nhân Số nguyên với Phép nhân Số thực trong DSP dấu phẩy tĩnh. 14](#_Toc164546269)

[1.2.2. Ex 2: Khối tính toán số liệu. 14](#_Toc164546270)

[Thiết lập bo mạch thí nghiệm 14](#_Toc164546271)

[1.2.3. Ex 3: Bộ nhớ. 19](#_Toc164546272)

[Thiết lập trên bo mạch thí nghiệm: 19](#_Toc164546273)

[Kiểm tra Chương trình 19](#_Toc164546274)

[1.2.4. Ex 4: Định địa chỉ. 25](#_Toc164546275)

[Thiết lập kết nối trên bo mạch: 25](#_Toc164546276)

[1.3. U3: Vào/Ra và các Ngoại vi. 30](#_Toc164546277)

[1.3.1. Ex 1: Ứng dụng với I/O và Ngoại vi. 30](#_Toc164546278)

[Thiết lập kết nối trên bo mạch thí nghiệm 31](#_Toc164546279)

[Các Thanh ghi Ngoại vi 31](#_Toc164546280)

[1.4. U4: Xử lý thời gian thực. 33](#_Toc164546281)

[1.4.1. Ex 1: Lấy mẫu và chuyển đổi ADC/DAC. 33](#_Toc164546282)

[Ứng dụng Lấy mẫu tín hiệu đầu vào 33](#_Toc164546283)

[Quan sát trong miền Tần số. 39](#_Toc164546284)

[1.4.2. Ex 2: Các ứng dụng FFT và Tối ưu. 42](#_Toc164546285)

[1.5. U5: Các ứng dụng xử lý tín hiệu 42](#_Toc164546286)

[1.5.1. Ex 1: Ứng dụng các bộ lọc FIR và IIR. 42](#_Toc164546287)

# GIỚI THIỆU

Tài liệu này ghi nhận các kết quả thí nghiệm của sinh viên dựa trên quá trình làm việc thực tế với bo mạch và các thiết bị thí nghiệm.

1. Thí nghiệm Xử lý tín hiệu - Digital Signal Processor.
   1. U1: Làm quen với hệ thống thí nghiệm DSP.
      1. Ex 1: Các kiến thức cơ bản.
      2. Ex 2: Tìm hiểu tổng quan về bo mạch thí nghiệm DSP.

Giới thiệu bo mạch Thí nghiệm

Làm quen với một số linh kiện, mạch điện và khối chức năng trong bo mạch thí nghiệm Xử lý tín hiệu số (DSP).

* + - * 1. Xác định vị trí các điểm nối đất chung trên bảng mạch DSP. Tắt nguồn, sau đó dùng ôm kế kiểm tra xem các điểm đất có được kết nối với nhau hay không.

có.

không.

|  |  |
| --- | --- |
| Chọn câu trả lời: | a |

* + - * 1. Bật công tắc cấp nguồn cho khối mạch DSP, các đèn trên Đế và bo mạch sẽ sáng.
        2. Sử dụng vôn kế DC, đo dải điện áp ở đầu ra của khối mạch nguồn dc. Để làm như vậy, hãy thay đổi chiết áp của nguồn dc từ giá trị nhỏ nhất đến giá trị lớn nhất của nó. Điện áp tối thiểu (Vmin) và điện áp tối đa (Vmax) từ nguồn DC là bao nhiêu?

|  |  |  |
| --- | --- | --- |
| Vmin = | -0.8 | V |
| Vmax= | 0.9 | V |

* + - * 1. Kết nối micrô với đầu vào của bộ khuếch đại trước micrô và kết nối đầu ra của bộ khuếch đại trước với đầu vào của bộ khuếch đại âm thanh.</p>

Có thể sử dụng tai nghe nếu cần thiết, cắm tai nghe vào ổ giắc trong khối mạch khuếch đại âm thanh.

* + - * 1. Trong khi nói vào micro, điều chỉnh các biến trở GAIN trong mạch tiền khuếch đại micro và mạch khuếch đại âm thanh..
        2. Tháo tất cả các dây dẫn có trên bảng mạch.

Làm quen với bảng mạch bằng chương trình DSP

Làm quen với phần mềm lập trình CCS. Dịch, nạp và chạy chương trình trên bo mạch DSP.

* + - * 1. Kết nối với máy tính bằng cổng USB số 2 như trong hình vẽ.

Đảm bảo rằng nguồn điện của Đế mạch đã được BẬT và phần mềm Code Composer trên máy tính được cài đặt như giải thích trong Phụ Lục C.

Chạy phần mềm Code Composer (CCS) trên máy tính.

* + - * 1. Kích hoạt dự án exercise\_1\_1 bằng cách nhấp chuột vào dự án đó trong cửa sổ Project Explorer.

Chạy Debug bằng cách click vào biểu tượng A picture containing text, microscope, engine, device

Description automatically generatedđể tải chương trình vào chip DSP.

Tiếp theo, nhấn vào biểu tượng  để chạy chương trình.

Kiểm tra thông tin hiển thị trên màn hình LCD của bo mạch DSP:

LabVolt 91031 Ex1-1

Echo Generator

* + - * 1. Cắm dây Microphone và kết nối đầu ra mạch tiền khuếch đại với đầu vào mạch khuếch đại âm thanh như hình minh hoạ.
        2. Đặt các công tắc DIP trong khối I/O Interface về vị trí 0 (ấn xuống dưới). Nhấn nút INT0 để màn hình LCD hiển thị “Echo Delay: 0ms”
        3. Nối dây trên bo mạch DSP như hình vẽ hướng dẫn: đầu ra mạch khuếch đại Microphone nối với đầu vào khối CODEC, đầu ra khối CODEC nối đến đầu vào Khuếch đại Audio.

Nói vào Micro, điều chỉnh chiết áp GAIN của bộ tiền khuếch đại và bộ khuếch đại âm thanh để thu được chất lượng âm thanh tốt ở đầu ra. Nhận xét về ảnh hưởng chức năng của các chiết áp.

Lưu ý: Dòng cuối của LCD hiển thị một dãy hình chữ nhật màu đen khi ta nói vào micro. Độ dài của dãy đó tương ứng với độ lớn của tín hiệu đầu vào.

Chương trình chạy trong DSP sẽ nhận tín hiệu đầu vào từ khối CODEC, hiển thị biên độ tín hiệu lên màn hình LCD và xử lý tín hiệu rồi gửi đến đấu ra của khối CODEC.

* + - * 1. Điều chỉnh công tắc DIP để đặt số nhị phân số 8-bit 15 (00001111) đặt vào DSP. Bit 1 ứng với việc ấn công tắc ở vị trí I.

Nhấn nút INT0 để hiển thị Echo Dela: 15ms. Chú ý đến ảnh hưởng của việc xử lý với âm thanh đầu ra

* + - * 1. Lặp lại bước 12 với các giá trị sau:

031 (0001 1111b)

063 (0011 1111b)

127 (0111 1111b)

255 (1111 1111b)

Tăng dần độ trễ, hiệu ứng vang sẽ thể hiện khá rõ. Thời gian tối thiểu cần thiết để nhận biệt được sự lặp lại là bao nhiêu (nhận biết hiệu ứng trễ):

Thường là 5 đến 15 mili giây

Thường là 20 đến 80 mili giây

Thường là 100 đến 160 mili giây

Thường là 180 đến 300 mili giây

* + - * 1. Click vào Shape, rectangle

           Description automatically generated để dừng chương trình.

Làm việc với ứng dụng thứ 2

Kết nối mạch thí nghiệm như ở phần trước, trong phần này ta sẽ thử nghiệm một chương trình DSP khác.

* + - * 1. Trong cửa sổ Project Explorer, kích chuột vào dự án exercise\_1\_1b. Chạy Debug bằng cách click vào  để tải chương trình vào mạch. Tiếp theo, click vào  để chạy chương trình. Thông tin hiển thị trên màn hình LCD của bảng mạch DSP như sau:
        2. Chạy ứng dụng và trải nghiệm. Nhấn vào các nút INT0/INT1 và sửa đổi giá trị của công tắc DIP. Nói vào micrô hoặc sử dụng các nguồn phát tín hiệu âm thanh. Chức năng nút INT0 trong ứng dụng này là:

Để chọn chế độ Giọng nói.

Để chọn chế độ Tiếng vang.

Để chọn chế độ Flanger.

Tất cả những điều trên.

|  |  |
| --- | --- |
| Chọn câu trả lời: | d |

Chức năng nút INT1 trong ứng dụng này là:

Để thay đổi chế độ.

Để cập nhật biến (giá trị được đọc từ công tắc DIP).

Để đặt lại công tắc DIP.

Tất cả những điều trên.

|  |  |
| --- | --- |
| Chọn câu trả lời: | b |

* + - * 1. Click vào biểu tượng Shape, rectangle

           Description automatically generated để kết thúc chương trình. Quay lại tab dự án và thoát khỏi Code Composer. TẮT nguồn điện và tháo các kết nối (dây dẫn) trên bảng mạch.
    1. Ex 3: Tìm hiểu Tool lập trình và cấu trúc Project.

Trình Phát triển và Cấu trúc Dự án

* + - * 1. Kết nối đầu ra của khối DC SOURCE với đầu vào ANALOG INPUT của CODEC và đầu ra ANALOG OUTPUT của CODEC với đầu vào bộ khuếch đại Audio như hình hướng dẫn.
        2. Kết nối với máy tính bằng cổng USB số 2 như trong hình vẽ. Đảm bảo rằng nguồn điện của Đế mạch đã được BẬT và phần mềm Code Composer trên máy tính được cài đặt như hướng dẫn trong Phụ Lục C. Chạy phần mềm Code Composer (CCS) trên máy tính.
        3. Kích hoạt dự án exercise\_1\_2 bằng cách nhấp chuột vào dự án đó trong cửa sổ Project Explorer.
        4. Chạy Debug bằng cách click vào biểu tượng  để tải chương trình vào chip DSP. Chưa kích lệnh chạy chương trình.
        5. Trên giao diện phần mềm CCS, địa chỉ bắt đầu của hàm Main được tìm thấy tại:

Trong cửa sổ Project Explorer.

Trong cửa sổ Debug ở cuối hàm main().

Trong cửa sổ Disassembly khi bắt đầu phiên gỡ lỗi.

Cả b và c.

|  |  |
| --- | --- |
| Chọn câu trả lời: | d |

* + - * 1. Chuyển đến tab Registers và tìm thanh ghi Bộ đếm chương trình (PC). Thanh ghi này nằm trong danh sách Core Registers. Giá trị của thanh ghi PC có giống với địa chỉ bắt đầu của hàm main hay không?

Có.

Không.

|  |  |
| --- | --- |
| Chọn câu trả lời: | a |

Ấn nút thực hiện lệnh  để chạy đến câu lệnh tiếp theo. Giá trị mới của thanh ghi PC sẽ là:

Địa chỉ tuần tự tiếp theo trong chương trình.

Địa chỉ bắt đầu của hàm chính.

Địa chỉ kết thúc hàm.

Không có điều nào ở trên.

|  |  |
| --- | --- |
| Chọn câu trả lời: | a |

Con số này tương ứng với:

Địa chỉ nơi chức năng chính bắt đầu.

Địa chỉ của lệnh hợp ngữ hiện tại.

Tổng số dòng chương trình.

Địa chỉ của lệnh hợp ngữ cuối cùng.

|  |  |
| --- | --- |
| Chọn câu trả lời: | b |

Chức năng của thanh ghi PC là:

Để theo dõi tổng số địa chỉ lệnh trong bộ nhớ.

Để theo dõi tổng số chương trình trong bộ nhớ.

Để theo dõi địa chỉ của lệnh hiện tại trong bộ nhớ.

Tất cả những điều trên.

|  |  |
| --- | --- |
| Chọn câu trả lời: | c |

* + - * 1. Trong khi theo dõi các tab main.c, Disassembly và Variables, hãy kích các loại lệnh Debug chạy từng bước khác nhau và nhận xét các thay đổi trong các biến số và cửa sổ trên.
        2. Ấn nút Run  để chạy liên tục chương trình trong chip DSP. Điều chỉnh chiết áp GAIN để đạt được mức âm lượng phù hợp ở đầu ra. Kiểm tra thông tin hiển thị trong LCD trên bo mạch, trong đó x là một con số cụ thể.
        3. Như vậy từ hiển thị trên LCD và âm thanh phát ra từ loa, ta nhận thấy chương trình DSP đang chạy là một ứng dụng phát tín hiệu hình sin. Điều chỉnh chiết áp trong khối DC SOURCE và theo dõi ảnh hưởng đến tín hiệu đầu ra, ta nhận thấy:

Tần số của tín hiệu hiển thị được sửa đổi khi điều chỉnh chiết áp.

Giá trị của tần số cũng được sửa đổi tương ứng trên màn hình LCD.

Khi tần số thay đổi thì tín hiệu âm thanh cũng thay đổi.

Tất cả những điều trên.

|  |  |
| --- | --- |
| Chọn câu trả lời: | d |

* + - * 1. Quan sát tín hiệu đầu ra trên kênh CH2 của máy hiện sóng. Biên độ của tín hiệu có bị ảnh hưởng khi tần số thay đổi không:

Có.

Không.

|  |  |
| --- | --- |
| Chọn câu trả lời: | b |

Tần số biến thiên trong khoảng nào:

Tần số có thể dao động từ khoảng 1 Hz đến 500 Hz ± 5%.

Tần số có thể dao động từ khoảng 50 Hz đến 50 kHz ± 5%.

Tần số có thể dao động từ khoảng 10 Hz đến 5027 Hz ± 5%.

|  |  |
| --- | --- |
| Chọn câu trả lời: | c |

* + - * 1. Kích vào biểu tượng Icon

           Description automatically generated để tạm dừng chương trình.
        2. Một phần quan trọng của gỡ lỗi (Debug) là có thể biết giá trị của một biến cụ thể tại một thời điểm nhất định khi chương trình hoạt động. Điều này có thể được thực hiện trong hầu hết các trường hợp bằng cách sử dụng tab Expressions hoặc Variables.

Nhưng trong trường hợp này thì việc sử dụng các Tab trên không thuận lợi. Trong file main.c của dự án Exercise 1-2, tại dòng 22 có một biến mảng được khai báo:

Int16 sintbl[2048]; Mảng này có tên là sintbl và chứa 2048 phần tử có kiểu số nguyên 16-bit. Từ dòng 47 đến 51 là các câu lệnh để tính giá trị từng phần tử trong bảng sintbl.

Mở tab Expressions và gõ sintbl vào mục new ta sẽ nhận được dữ liệu của mảng. Khi muốn xem đủ các phần tử thì cần mở rộng khung hình. Như vậy phương pháp này rõ ràng không phù hợp để theo dõi các biến mảng có số phần tử lớn:

|  |  |
| --- | --- |
| Ghi lại địa chỉ của biến mảng sintbl: 0x | 0x001862 |

* + - * 1. Một phương pháp khác để theo dõi bảng dữ liệu hiệu quả hơn, đó là sử dụng cửa sổ Memory. Vào menu View trong giao diện của CCS và chọn Memory Browser. Xác định địa chỉ mảng sintbl từ tab Expression, nhập giá trị này vào cửa sổ Memory, chọn loại dữ liệu DATA rồi ấn Enter. Chọn kiểu dữ liệu là số nguyên 16 bit có dấu.

Các dữ liệu của mảng được hiển thị như hình sau.

Đây là một cải tiến, nhưng với kích thước quá lớn của mảng sintbl, ta còn một phương pháp khác để quan sát dữ liệu hiệu quả hơn.

* + - * 1. Sử dụng công cụ đồ họa là một phương pháp để quan sát mảng dữ liệu kích thước lớn.
        2. Trong menu Tools, chọn Graph, rồi chọn Single Time. Trong hộp thoại thiết lập các thông số như hình và nhấn OK. Địa chỉ Start Address là vị trí trong bộ nhớ mà mảng dữ liệu bắt đầu. Biến sintbl chứa địa chỉ đó.

Hình đồ thị nào sau đây thể hiện tốt nhất dữ liệu của bảng sintbl: a / b

|  |  |
| --- | --- |
| Chọn câu trả lời: | a |

* + - * 1. Sử dụng các điểm ngừng (breakpoints) là giải pháp hết sức quan trọng trong debug các tiến trình hoạt động của Chip. Trong chế độ Debug và DSP ở trạng thái nghỉ, đặt một điểm ngừng tại vị trí //Breakpoint #01 bằng cách kích đúp chuột vào số thứ tự dòng lệnh tương ứng.

Theo dõi giá trị biến freq trong tab expressions.

Nhấn nút chạy  để chạy chương trình. Việc thực thi chương trình sẽ dừng khá nhanh ở dòng thích hợp trong hàm main. Chú ý giá trị của biến tần số:

|  |  |
| --- | --- |
| Biến tần số quan sát được (Ghi kèm Hz ): | 5027.344 |

Điều chỉnh chiết áp nguồn dc theo bất kỳ hướng nào và nhấn lại nút chạy. Quá trình sẽ chạy trong thời gian ngắn cho đến khi nó dừng lại ở cùng một dòng mã.

Giá trị hiện tại của biến freq có giống hoặc gần giống với giá trị đã quan sát ở bước trước không?

Có.

Không.

|  |  |
| --- | --- |
| Chọn câu trả lời: | a |

Nhấn bước màu vàng Icon

Description automatically generated with low confidence để thực thi dòng lệnh hiện tại.

Giá trị hiện tại của biến freq như thế nào?

Giá trị không đổi.

Giá trị được cập nhật tương ứng với vị trí mới của chiết áp nguồn dc.

Giá trị đã giảm xuống 0.

Tất cả đều sai.

|  |  |
| --- | --- |
| Chọn câu trả lời: | b |

* + - * 1. Điểm dừng được đặt trong code C bên dưới nhãn //Breakpoint #01 tương ứng với điểm dừng được đặt tự động trong cửa sổ Disassembly tại địa chỉ 0x023059.

Thêm điểm dừng trong cửa sổ Disassembly tại địa chỉ 0x023086. Điều này thêm một điểm dừng tương ứng trong tệp main.c ở dòng nào?

|  |  |
| --- | --- |
| Giá trị dòng: | 72 |

Điểm dừng tại dòng:

|  |  |
| --- | --- |
| Giá trị dòng: | 72 |

Xóa điểm dừng tại địa chỉ 0x023086, sau đó thêm điểm dừng trong tệp main.c tại dòng bạn vừa viết. Điểm dừng được thêm vào ở địa chỉ nào trong cửa sổ Disassembly?

0x023086

0x023049

0x023059

0x02307C

|  |  |
| --- | --- |
| Chọn câu trả lời: | d |

Lệnh tại địa chỉ 0x023046 có phải là phần đầu của lệnh tương ứng trong C không?

Yes

No

|  |  |
| --- | --- |
| Chọn câu trả lời: | b |

Tại sao:

Mã 0x023086 là bước trung gian của lệnh, không phải bắt đầu của lệnh (0x02307C).

Mã tại địa chỉ 0x023046 là một câu lệnh include.

Mã tại địa chỉ 0x023046 chỉ là một chú thích.

|  |  |
| --- | --- |
| Chọn câu trả lời: | a |

Xóa tất cả các điểm dừng đã chèn trong chương trình.

Sửa dữ liệu trong Bộ nhớ và Thanh ghi.

* + - * 1. Mở tệp chương trình ngắt interrupt.c.

Đặt điểm dừng sau chỉ dẫn //Breakpoint #02.

Quan sát rằng biến index được thay đổi bởi dòng lệnh này.

Mở tab Expressions và thêm biến index vào danh sách các biến cần theo dõi.

Có thể chỉnh sửa giá trị của biến index từ tab Expressions. Bấm vào trường Giá trị và thay đổi giá trị của biến.

* + - * 1. Cũng có thể chỉnh sửa các thanh ghi của DSP. Mở tab Register (Thanh ghi) và tìm thanh ghi XAR1 trong Core Registers(Thanh ghi lõi). Nhấp vào trường giá trị và chỉnh sửa giá trị của thanh ghi (theo hệ thập lục phân).

Xóa điểm ngắt và chạy lại trình gỡ lỗi bằng cách nhấn nút chạy . Chương trình phải chạy bình thường.

* + - * 1. Bấm dừng chương trình Icon

           Description automatically generated và chỉnh sửa giá trị thanh ghi PC trong tab Registers (Core Registers) thành: 0x023129.

Chạy lại chương trình . Chương trình có hoạt động lại bình thường không?

Có

Không

|  |  |
| --- | --- |
| Chọn câu trả lời: | b |

* + - * 1. Nhấp vào biểu tượng Shape, rectangle

           Description automatically generated để dừng chương trình. Quay lại tab Projects và thoát khỏi Code Composer.
        2. TẮT nguồn điện và tháo các kết nối (dây dẫn) trên bảng mạch.
  1. U2: Kiến trúc DSP.
     1. Ex 1: Bộ xử lý số học.

Biểu diễn số thập phân có dấu ở dạng Q14

* + - * 1. Chuyển đổi số -0.984375 sang biểu diễn ở dạng nhị phân Q14. Để thực hiện, ta cần mở phần mềm Calculator có sẵn trong Windows hoặc sử dụng một máy tính bỏ túi có chức năng chuyển đổi cơ số.
        2. Dùng máy tính để nhân -0,984375 với 2<sup>14</sup>. Điều này chia tỷ lệ giá trị thập phân phân số thành giá trị thập phân nguyên và giúp dễ dàng chuyển đổi số này thành số nhị phân (ở định dạng 2s).

|  |  |
| --- | --- |
| Số nguyên thu được là:? | -16128 |

* + - * 1. Sử dụng công cụ chuyển đổi thập phân sang nhị phân của máy tính của bạn để thu được giá trị nhị phân một Word (16 bit). Trên calculator kích chọn hiển thị dữ liệu 16 bit.

Nhập chuỗi bit nhị phân, lưu ý dấu cách sau mỗi 4 bit:?

|  |
| --- |
| 1100 0001 0000 0000 |

Nhập giá trị khi số trên là mã hoá bù 2:?

|  |
| --- |
| -16128 |

Chuỗi bit trên tương ứng với giá trị nào nếu nó là định dạng Q14):?

|  |
| --- |
| -0.984375 |

|  |  |  |
| --- | --- | --- |
| Chép lại chuỗi nhị phân nhận được: | 1100 0001 0000 0000 |  |

* + - * 1. Xác minh số nhị phân trên bằng giá trị thập phân -0.984375. Sử dụng trọng số định dạng Q14 để tính giá trị thập phân. Ghi kết quả nhận được vào ô sau:

|  |  |  |
| --- | --- | --- |
| Kết quả: | -0.984375 | 1100 0001 0000 0000 |

Chuyển số Nhị phân sang Thập phân

Chuyển đổi số nhị phân 1011 0000 1001 0011b (B093h) thành số thập phân với các dạng thức biểu diễn khác nhau.

* + - * 1. Số thập lục phân biểu diễn một “ số nguyên không dấu”. Xác định trọng số bit liên quan đến biểu diễn này:?

- 215 214 213 212 211 210 29 28 27 26 25 24 23 22 21 20.

215 214 213 212 211 210 29 28 27 26 25 24 23 22 21 20.

|  |  |
| --- | --- |
| Kết quả: | b |

Tính giá trị thập phân thu được và chọn kết quả đúng:?

215 + 213 + 212 + 27 + 24 + 21+ 20 = 45203

215 + 214 + 212 + 27 + 26 + 21+ 20 = 53443

215 + 213 + 212 + 28 + 24 + 21+ 20 = 45331

|  |  |
| --- | --- |
| Kết quả: | a |

Với: B093h = 1011 0000 1001 0011b

* + - * 1. Số thập lục phân được viết theo định dạng 2s. Xác định trọng số bit liên quan đến biểu diễn này:?

- 215 214 213 212 211 210 29 28 27 26 25 24 23 22 21 20.

215 214 213 212 211 210 29 28 27 26 25 24 23 22 21 20.

|  |  |
| --- | --- |
| Kết quả: | a |

Tính giá trị thập phân thu được và chọn kết quả đúng:?

-215 + 212 + 211 + 27 + 24 + 21+ 20 = -26477

-215 + 214 + 212 + 27 + 26 + 21+ 20 = -12093

-215 + 213 + 212 + 27 + 24 + 21+ 20 = -20333

|  |  |
| --- | --- |
| Kết quả: | c |

Với: B093h = 1011 0000 1001 0011b

* + - * 1. Số thập lục phân được viết bằng định dạng Q15. Xác định trọng số bit liên quan đến biểu diễn này:?

-20 2-1 2-2 2-3 2-4 2-5 2-6 2-7 2-8 2-9 2-10 2-11 2-12 2-13 2-14 2-15

-21 -20 2-1 2-2 2-3 2-4 2-5 2-6 2-7 2-8 2-9 2-10 2-11 2-12 2-13 2-14

-22 -22 -20 2-1 2-2 2-3 2-4 2-5 2-6 2-7 2-8 2-9 2-10 2-11 2-12 2-13

|  |  |
| --- | --- |
| Kết quả: | a |

Tính giá trị thập phân thu được và chọn kết quả đúng:?

-20 + 2-1 + 2-3 + 2-8 + 2-11 + 2-14 + 2-15 = -0.37051391

-20 + 2-2 + 2-3 + 2-8 + 2-11 + 2-14 + 2-15 = -0.62051391

-20 + 2-2 + 2-6 + 2-8 + 2-11 + 2-14 + 2-15 = -0.72988891

|  |  |
| --- | --- |
| Kết quả: | b |

Với: B093h = 1011 0000 1001 0011b

* + - * 1. Số thập lục phân được viết bằng định dạng dấu phẩy động 16 bit. Trong các bước tiếp theo, ta sẽ xác định trọng số bit liên quan đến biểu diễn này và tính giá trị thập phân thu được.

Hãy nhớ thêm vào bit ngầm định, các bit thể hiện giá trị của phần định trị là:?

1 0100 1111 0110

1 1011 0000 1001

1011 0000 1001 1

|  |  |
| --- | --- |
| Kết quả: | b |

Xác định trọng số bit liên quan đến phần định trị:?

-21 20 2-1 2-2 2-3 2-4 2-5 2-6 2-7 2-8 2-9 2-10 2-11

211 210 29 28 27 26 25 24 23 22 21 20 2-1

|  |  |
| --- | --- |
| Kết quả: | a |

Xác định trọng số bit liên quan đến phần mũ:?

20 21 22 23

-23 22 21 20

|  |  |
| --- | --- |
| Kết quả: | b |

Tính giá trị phần định trị:?

20 + 2-2 + 2-3 + 2-8 + 2-11 = 1.37939453125

-21 + 20 + 2-2 + 2-3 + 2-8 + 2-11 = -0.62060546875

|  |  |
| --- | --- |
| Kết quả: | b |

Xác định trọng số bit của phần mũ và tính giá trị của luỹ thừa:

21 + 20 = 3

22 + 21 = 5

22 + 20 = 4

|  |  |
| --- | --- |
| Kết quả: | a |

Giá trị thập phân của số ở định dạng dấu phẩy động 16 bit là gì:?

1.37939453125 x 23 = 4.13818359375

0.62060546875 x 23 = 4.96484375

-0.62060546875 x 23 = -4.96484375

|  |  |
| --- | --- |
| Kết quả: | c |

* + - * 1. Kết nối Bo mạch DSP với máy tính bằng cổng USB số 2. Đảm bảo rằng nguồn điện của Đế mạch đã được BẬT và phần mềm Code Composer trên máy tính được cài đặt như giải thích trong Phụ Lục C. Chạy phần mềm Code Composer (CCS) trên máy tính.
        2. Kích hoạt dự án exercise\_2\_1 bằng cách nhấp chuột vào dự án đó trong cửa sổ Project Explorer.

Chạy Debug bằng cách click vào biểu tượng A picture containing text, microscope, engine, device

Description automatically generated để tải chương trình vào chip DSP.

* + - * 1. Nhấn vào biểu tượng  để chạy chương trình. Kiểm tra thông tin hiển thị trên màn hình LCD của bo mạch DSP: và một số thông tin về chuyển đổi số.

Nhấn vào biểu tượng Icon

Description automatically generated để tạm dừng chương trình.

* + - * 1. Mở tab Expressions và gõ “var1” vào trường “new” để xem giá trị của biến này. Sử dụng tùy chọn Q-Values trong menu con Q-Values để chỉ định hiển thị giá trị ở định dạng Q. Nhập giá trị của biến là -0,984375.

Bây giờ, nhấp chuột phải vào giá trị và chọn định dạng decimal trong menu con Number Format. Giá trị này có trùng với giá trị thu được ở bước 2 không: Chọn a/b:

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Bây giờ, nhấp chuột phải vào biến “var1và chọn định dạng Decimal trong menu Number Format. Giá trị này có trùng với giá trị thu được ở bước 3 không?

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Chọn dạng thập lục phân (Hex) cho biến “var1” gõ 0xB093 để chỉnh sửa giá trị biến. Thay đổi định dạng của biến “var1” thành decimal (số thập phân). Số thập phân nhận được là:?

-16128

-20333

-20477

|  |  |
| --- | --- |
| Kết quả: | b |

Giá trị này có trùng với giá trị thu được ở bước 6 không:? Chọn a/b

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Giá trị Q15 của số này là bao nhiêu, thay đổi định dạng của var1 thành Q15. Kết quả là: Chọn a/b/c.

-16128

-20333

-0.620514

|  |  |
| --- | --- |
| Kết quả: | c |

Giá trị này có trùng với giá trị thu được ở bước 7 không:? Chọn a/b

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Ép kiểu cho biến “var1” thành kiểu float bởi lệnh Cast To Type. Nhập "float" vào cửa sổ cast to type và nhấn OK. Giá trị này có trùng với giá trị thu được ở bước 8 không:? Chọn a/b.

|  |  |
| --- | --- |
| Kết quả: | b |

Kích thước ban đầu của biến “var1” là: Chọn a/b

var1 thuộc loại số nguyên (32 bit) - kiểu interger 32 bit.

var1 thuộc loại số nguyên (16 bit) - kiểu interger 16 bit.

|  |  |
| --- | --- |
| Kết quả: | b |

Khi một biến đã được khai báo, ví dụ như biến “var3” khai báo kiểu float, sử dụng biểu diễn dấu phẩy động (32-bit), thì biến đó không thể biểu diễn dưới dạng thức khác.

Ví dụ một biến đã được khai báo dưới dạng float thì không thể hiển thị dưới dạng số định dạng Q. Tương tự, một biến được khai báo là một số nguyên không thể sử dụng biểu diễn dấu phẩy động.

Để kiểm tra giả định trên, ta thêm biến “var3” vào danh mục trong tab expressions và theo dõi.

* + - * 1. Đổi kiểu biến “var1” thành kiểu số nguyên.

Lưu ý biến “var1” không thể được dùng như một số nguyên không dấu và chứa giá trị như trong bước 5. Lý do là biến đã được khai báo kiểm số nguyên có dấu 16 bit.

Thêm biến varUnsigned vào Tab Expressions.

Sửa giá trị của nó thành 0xB093 và hiển thị kết quả ở dạng ký hiệu thập phân.

Giá trị này có trùng với giá trị thu được ở bước 5 không: Chọn a/b

|  |  |
| --- | --- |
| Kết quả: | a |

Kiểu của biến varUnsigned là gì:? Chọn a/b/c.

|  |  |
| --- | --- |
| Kết quả: | b |

So sánh phép nhân Số nguyên với Phép nhân Số thực trong DSP dấu phẩy tĩnh.

Trong file main.c đặt 2 điểm dừng bên dưới dòng Comment //Breakpoint #01 và //Breakpoint #02.

* + - * 1. Nhấn nút run  để chạy chương trình, nó sẽ dừng lại ở điểm dừng thứ nhất .

Dòng lệnh bên dưới //Breakpoint #01 thực hiện nhân một số thực là biến var3 với chính nó.

Thực hiện chạy từng lệnh bằng cách kích vào biểu tượng . Quan sát các lệnh trong cửa sổ Disassembly và kích vào biểu tượng  một vài lần. Ta thấy phải thực hiện khá nhiều lệnh Assembly cho một lệnh nhân số thực trong C.

* + - * 1. Nhấn nút run  để chạy lại chương trình cho đến khi nó dừng lại ở dòng bên dưới chỉ dẫn //Breakpoint #02

Dòng lệnh bên dưới thực hiện nhân một số nguyên là biến var1với chính nó.

Quan sát các lệnh trong cửa sổ Disassembly và kích vào biểu tượng  một vài lần. Nhận thấy chỉ có vài lệnh Assembly cho phép nhân số nguyên (4 lệnh).

Minh chứng trên cho thấy sự phức tạp thuật toán và thời gian xử lý khi thực hiện phép nhân số thực trên chip DSP dấu phẩy tĩnh.

* + - * 1. Loại bỏ tất cả các điểm dừng breakpoints trong file. Click vào biểu tượng Shape, rectangle

           Description automatically generated để kết thúc chương trình. Quay lại tab projects và thoát khỏi Code Composer.

Tắt nguồn điện và tháo các kết nối (dây dẫn) trên bảng mạch.

* + 1. Ex 2: Khối tính toán số liệu.

Thiết lập bo mạch thí nghiệm

* + - * 1. Kết nối Bo mạch DSP với máy tính bằng cổng USB số 2. Đảm bảo rằng nguồn điện của Đế mạch đã được BẬT và phần mềm Code Composer trên máy tính được cài đặt như giải thích trong Phụ Lục C. Chạy phần mềm Code Composer (CCS) trên máy tính.
        2. Kích hoạt dự án exercise\_2\_2 bằng cách nhấp chuột vào dự án đó trong cửa sổ Project Explorer. Mở đồng thời hai file main.c và LVex2\_2.asm. Chạy Debug bằng cách click vào biểu tượng A picture containing text, microscope, engine, device

           Description automatically generated để tải chương trình vào chip DSP.
        3. Nhấn vào biểu tượng  để chạy chương trình. Kiểm tra thông tin hiển thị trên màn hình LCD của bo mạch DSP:

Festo 91031 Ex2-2

* + - * 1. Điều chỉnh chuyển mạch DIP trong khối I/O theo giá trị bất kỳ ta chọn. Quan sát kết quả trên LCD. Kiểm tra hai chữ số cuối trong thanh ghi AC0 có liên quan với giá trị chuyển mạch DIP hay không. Kích vào nút Icon

           Description automatically generated để tạm dừng chương trình.

Trong file LVex2\_2.asm xác định điểm đánh dấu MARKER1.

Giá trị thêm vào thanh ghi tích lũy AC0 có liên quan với giá trị của chuyển mạch DIP. Giá trị ban đầu là 8111h được dịch sang trái 15 bit.

Giá trị hiển thị trên LCD sẽ là bao nhiêu nếu thay lệnh: add #8111h<<#15,AC0 bằng lệnh add #8111h<<#8,AC0 với chuyển mạch DIP đặt là 1111 0111b:

0x00811B3

0x00811FF

0x00811F7

|  |  |
| --- | --- |
| Kết quả: | c |

* + - * 1. Tạm dừng thực thi mã lệnh DSP, chèn một điểm ngắt breakpoint vào file LVex2\_2.asm, ngay sau nhãn Operations . Lưu ý, ngay trước điểm dừng breakpoint trên là lệnh không kích hoạt chế độ sign-extension mode: bclr SXMD;no sign extension.

Các giá trị trong bảng sau được dùng với dạng số bù 2

* + - * 1. Mở cửa sổ Registers và chọn các thanh ghi AC0HL, T0, ST0 trong mục Core registers. Ấn nút  để khởi chạy chương trình trên board.

Quan sát các dòng lệnh trong file LVex2\_2.asm và kích vào lệnh  để chạy từng bước. Quan sát giá trị các thanh ghi trên để nhận biết kết quả từ các lệnh ADD, SUB, AND, OR, NOT, MACM, MPYM. Ghi lại giá trị (của thanh ghi AC0) vào hộp thoại sau, ứng với từng bước thực hiện:

|  |  |  |
| --- | --- | --- |
| TT | Lệnh | Giá trị AC0: |
| 1 | ADD (x, y): | 0x000000000D1 |
| 2 | SUB (x, y): | 0x00000000097 |
| 3 | AND (x, y): | 0x00000000014 |
| 4 | OR: (x, y): | 0x000000000BD |
| 5 | NOT (x): | 0xFFFFFFFFE2 |
| 6 | MACM (AC0 z + x . y): | 0x0000001465 |
| 7 | MPYM (AC0 = x . y): | 0x0000001464 |
| 8 | MACM (AC0 = AC00 + x . -y) | 0x0000000000 |
| 9 | ADD (17,w): | 0x0000010000 |
| 10 | MPYM(z=mul1 . mul2): | 0xFFFFFFDE11 |

* + - * 1. Quay trở lại điểm dừng và lệnh loại bỏ chế độ sign extension. Kích hoạt chế độ "sign extension" bằng cách không cho lệnh này thực hiện - chuyển lệnh thành Comment với dấu ";" đặt trước: ;bclr SXMD ;no sign extension.

Kích nút Shape, rectangle

Description automatically generated để ngừng chế độ Debug rồi chạy lại bằng nút A picture containing text, microscope, engine, device

Description automatically generated. Điều này sẽ dịch lại chương trình và nạp lại mã lệnh mới vào DSP, lúc này chế độ sign extension đã kích hoạt.

Ấn nút run  để chạy chương trình. Lúc này chương trình sẽ dừng tại điểm ngắt breakpoint.

* + - * 1. Thực hiện lại các lệnh chạy từng bước (với lệnh ) và ghi nhận lại kết quả thanh ghi ở lệnh ADD (17,w):

|  |  |
| --- | --- |
| Kết quả: ADD (17,w): Giá trị AC0 = | 0x00000000000 |

Giá trị của w thêm vào thanh ghi AC0 là: (dòng thực hiện trước lệnh cộng Add)?

|  |  |
| --- | --- |
| Kết quả: w = | 0xFFFFFFFFEF |

Ghi nhận kết quả từ bước trước:

|  |  |
| --- | --- |
| Bước 8. ADD (17, w). Giá trị AC0 = | 0x00000000000 |
| Bước 6. ADD (17, w). Giá trị AC0 = | 0x0000010000 |

So sánh kết lệnh cộng ADD (17, w) với giá trị ở bước 6, chế độ "sign extension" có cần thiết không:

Chế độ "Sign extension" cho kết quả đúng trong trường hợp này (số âm không bị nhầm với số dương) và rất cần thiết.

Chế độ "Sign extension" không ảnh hưởng đến kết quả tính.

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Loại bỏ điểm dừng đã đặt ở trong file LVex2\_2.asm, sau nhãn Operations.

Các bit cờ Tràn, Chế độ bão hoà và Nhớ, Overflow.

* + - * 1. Dừng chương trình bởi kích nút để đọc được các giá trị bit. Chuyển đến nhãn MARKER3 trong file LVex2\_2.asm và đặt một điểm dừng ở lệnh MOV #0,AC0 và một điểm dừng ở lệnh bclr CARRY (ngay sau lệnh MAC) như hình vẽ.

Quan sát thấy chế độ "saturation mode" không được kích hoạt (lệnh bset SATD bị vô hiệu bởi dấu comment ";").

Đoạn code sau nhân biến \_bigV (giá trị cực đại của một số nguyên 2 byte là = 7FFFh) với chính nó và đặt kết quả vào thanh ghi AC0. Dòng mã RPT #257 lặp lại phép nhân trên 257 lần, kết quả là tạo ra một số rất lớn đặt trong thanh ghi AC0.

* + - * 1. Ấn nút  để chạy, chương trình sẽ dừng lại tại điểm ngắt đầu tiên. Thực hiện lệnh từng bước (với nút lệnh ) và quan sát giá trị thanh ghi AC0 tăng lên sau mỗi lệnh nhân MAC.

Ấn nút  một lần nữa, chương trình sẽ thực hiện đủ 257 phép nhân và dừng ở điểm ngắt thứ 2. Ghi lại các giá trị thanh ghi: AC0, ACOV0:

|  |  |
| --- | --- |
| AC0 = | 0x80FDFC0204 |
| ACOV0= | 1 |

Các lệnh nhân có làm cho thanh ghi AC0 bị tràn không? Chọn a/b

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Thực hiện lệnh bclr CARRY và bclr ACOV0 (kích vào biểu tượng ) để xóa (đặt giá trị về 0) các bit cờ CARRY và bit Cờ tràn AC0 trong thanh ghi trạng thái ST0 status.

Tiếp theo, thực hiện từng lệnh và ghi lại giá trị bit CARRY ở các lệnh sau:

|  |  |  |
| --- | --- | --- |
| TT | Lệnh | Giá trị CARRY: |
| 1 | SUB #1001h,AC0 | 0 |
| 2 | ADD #1001h,AC0 | 1 |
| 3 | SUB #1001h,AC0 | 1 |
| 4 | ADD #1001h,AC0 | 0 |

Bit CARRY có giá trị như dự đoán không? Chọn a/b

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Ấn nút tạm dừng chương trình, trong file LVex2\_2.asm đặt điểm ngừng tại nhãn Operations. Chú ý lệnh bclr SXMD được thực thi (không có ";" trước lệnh này) để không kích hoạt chế độ sign-extension (chế độ mở rộng bit dấu).

Tiếp theo, xóa dấu ";" trước lệnh: ;bset SATD;Saturation mode bit for the D-Unit thành bset SATD để đặt chế độ Saturation (tạo cờ trạng thái bão hòa).

Kích vào nút Shape, rectangle

Description automatically generated để kết thúc chương trình, rồi kích lại vào nút A picture containing text, microscope, engine, device

Description automatically generated lần nữa để dịch và nạp lại chương trình vào DSP, lúc này DSP đã được kích hoạt chế độ "overflow saturation" - tạo cờ tràn và bão hòa.

Ấn vào nút run  để chạy lại chương trình. Ngay sau đó chương trình sẽ dừng ở điểm ngắt thứ nhất.

* + - * 1. Ấn vào nút run  lần thứ 2 để chương trình dừng tại điểm ngắt cuối cùng sau khi đã hoàn thành các lệnh nhân. Ghi lại các giá trị AC0, AC0V0:

|  |  |
| --- | --- |
| AC0 = | 0x7FFFFFFFFF |
| ACOV0= | 0 |

Các lệnh nhân có làm cho thanh ghi AC0 bị tràn không? Chọn a/b

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Kích vào nút Shape, rectangle

           Description automatically generated để dừng chương trình. Loại bbỏ mọi điểm ngắt. Đặt dấu (;) vào trước lệnh bset SATD để loại bỏ tác dụng của nó.

Loại bỏ dấu (;) trước lệnh bclr SXMD ở ngay trước nhãn Operations.

* + - * 1. Kích hoạt dự án exercise\_2\_2b bởi kích chuột vào dự án này trong cửa sổ Project Explorer.

Mở đồng thời 2 file main.c và LVex2\_2b.asm cuả dự án exercise\_2\_2b.

* + - * 1. Quan sát nội dung của file LVex2\_2b.asm. Ta thấy có 2 đoạn chương trình tính toán đầu ra của một bộ lọc, đây là một tác vụ cơ bản của DSP.

Đoạn chương trình thứ nhất không sử dụng cơ chế tính toán song song. Nó sử dụng lệnh nhân và gọi nhiều lần trong vòng lặp.

Thời gian thực thi của giải thuật này nằm trong biến delta1, đã được khai báo trong file main.c. Giá trị của biến này là hiệu của hai bộ đếm (timeCnt1 và timeCnt2).

Đoạn chương trình thứ hai làm cùng nhiệm vụ chức năng như đoạn thứ nhất, nhưng nó sử dụng cơ chế tính toán song song.

Có thể thấy đó chính là một cặp lệnh MAC được phân cách bởi dấu (::).

Thời gian thực thi của giải thuật này nằm trong biến delta2, đã được khai báo trong file main.c. Giá trị của biến này là hiệu của hai bộ đếm (timeCnt3 và timeCnt4).

* + - * 1. Trong cửa sổ Expressions, thêm hai biến: delta1 và delta2.Lưu ý bộ đếm trong các đoạn chương trình trên giảm 1 sau mỗi 2 xung đồng hồ. Như vậy số xung đồng hồ cần cho mỗi giải thuật là delta x 2;
        2. Vào chế độ Debug bởi kích vào nút Shape, rectangle

           Description automatically generated và load chương trình vào DSP. Thêm một điểm ngắt vào file main.c tại dòng đánh dấu //Breakpoint #01.

Điều này cho phép tạm dừng chương trình trước khi thực hiện các đoạn code trong dự án ex2\_2b.

* + - * 1. Kích chuột vào nút để chạy chương trình trong DSP. Lúc này, ta có thể nhìn thấy thông tin hiển thị trên LCD:

L.V. 91031 Ex2-2b

Memory &amp; Profiling

* + - * 1. Các đoạn chương trình trong ex2\_2b đã được thực hiện. Ấn nút chạy từng bước  hai lần để tính toán các giá trị delta1 và delta2 rồi nhập số liệu vào các ô sau.

|  |  |  |  |
| --- | --- | --- | --- |
| delta1 = | 196 | Tương ứng với số xung đồng hồ = | 392 |
| delta2 = | 113 | Tương ứng với số xung đồng hồ = | 226 |

Cơ chế tính toán song song có làm tăng dung lượng xử lý của DSP không? Chọn a/b

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Click vào biểu tượng Shape, rectangle

           Description automatically generated để kết thúc chương trình. Quay lại tab projects và thoát khỏi Code Composer. Tắt nguồn điện và tháo các kết nối (dây dẫn) trên bảng mạch.
    1. Ex 3: Bộ nhớ.

Thiết lập trên bo mạch thí nghiệm:

* + - * 1. Thực hiện các kết nối trên bo mạch như hình vẽ. Có thể đưa tín hiệu audio từ máy tính đến đầu vào của mạch tiền khuếch đại Microphone, hoặc sử dụng Microphone.
        2. Kết nối Bo mạch DSP với máy tính bằng cổng USB số 2. Đảm bảo rằng nguồn điện của Đế mạch đã được Bật và phần mềm Code Composer trên máy tính được cài đặt như giải thích trong Phụ Lục C. Chạy phần mềm Code Composer (CCS) trên máy tính.
        3. Kích hoạt dự án exercise\_2\_3 bằng cách nhấp chuột vào dự án đó trong cửa sổ Project Explorer.

Mở đồng thời hai file main.c và interrupt.c.

* + - * 1. Chạy Debug bằng cách click vào biểu tượng A picture containing text, microscope, engine, device

           Description automatically generated để tải chương trình vào chip DSP.

Nhấn vào biểu tượng để chạy chương trình. Kiểm tra thông tin hiển thị trên màn hình LCD của bo mạch DSP:

LabVolt 91031 Ex2-3

Playback/Recorder

Kiểm tra Chương trình

Chương trình này điều khiển chip DSP ghi lại tín hiệu Audio đầu vào với nhiều phương pháp lấy mẫu khác nhau.

Các chế độ được lựa chọn bằng chuyển mạch DIP trên phần giao diện I/O.

Đặt 3 bit cuối của chuyển mạch DIP giá trị từ 0 đến 4 để thực hiện các chế độ lấy mẫu:

|  |  |  |
| --- | --- | --- |
| 0 | (000b) | Lấy mẫu 16-bit |
| 1 | (001b) | Lấy mẫu 8-bit sử dụng luật nén μ-law |
| 2 | (010b) | Lấy mẫu 8-bit với chế độ cắt ngắn mẫu |
| 3 | (011b) | Lấy mẫu 16-bit, lưu trong bộ nhớ flash |
| 4 | (100b) | Chế độ dùng để xóa bộ nhớ flash |

Hai nút ấn trên bo mạch dùng để:</p>

INT0: Bắt đầu ghi âm hoặc để xóa bộ nhớ flash(trong chế độ 4).

INT1: Để phát lại âm thanh đã ghi

* + - * 1. Đặt giá trị của chuyển mạch DIP trong khối I/O về 0 (mọi bit đặt ở giá trị 0 - ấn xuống).
        2. Ấn phím INT0 trên bo mạch để kích hoạt chế độ ghi âm. Sử dụng microphone để đưa tín hiệu vào DSP. Hoặc có thể dùng máy tính phát nhạc, lúc này cần chỉnh hệ số khuếch đại của khối tiền khuếch đại ở mức thấp.

Điều chỉnh chiết áp GAIN trong khối tiền khuếch đại để tránh tình trạng âm thanh bão hòa.

Khi nói vào microphone, chú ý đến chuỗi hiển thị ở dòng dưới cùng trên LCD. Độ dài chuỗi ứng với cường đội tín hiệu từ đầu vào đến từ mạch tiền khuếch đại. Khi thấy chuỗi sáng này hiển thị trên toàn bộ dòng một cách liên tục thì cần điều chỉnh lại chiết áp GAIN ở mức thấp để tránh tình trạng bão hòa.

NOTE: Có thể lặp lại bước này và bước trước một vài lần để có mức tín hiệu và hệ số khuếch đại hợp lý.

* + - * 1. Tiếp theo, ấn phím INT1 để phát lại âm thanh đã ghi. Điều chỉnh chiết áp GAIN của của mạch khuếch đại Audio để đạt được mức tín hiệu phù hợp ở đầu ra Loa hoặc tai nghe.
        2. Điều chỉnh chuyển mạch DIP trong khối I/O về giá trị 0 để kích hoạt chế độ 16-bit.
        3. Nói vào Micro hoặc phát nhạc từ máy tính rồi ấn phím ghi INT0. Khoảng thời gian dài nhất ghi được là?: Chọn a/b/c

Sấp xỉ 1 giây

Sấp xỉ 2 giây

Sấp xỉ 4 giây

|  |  |
| --- | --- |
| Kết quả: | c |

* + - * 1. Phát lại tín hiệu Audio INT1. Đánh giá chất lượng tín hiệu ghi được?: Chọn a/b/c/d

Phụ thuộc vào chất lượng tín hiệu gốc và thiết lập các chiết áp GAIN.

Tốt hơn với chế độ 8-bit.

Âm thanh về cơ bản là tốt.

Tất cả các điều trên đều đúng.

|  |  |
| --- | --- |
| Kết quả: | d |

* + - * 1. Tín hiệu được lấy mẫu và lưu trữ trong một bảng tên là recBuff. Kích vào nút Icon

           Description automatically generated để tạm dừng chương trình. Mở cửa sổ Expressions và thêm biến recBuff vào danh sách. Số lượng mẫu trong biến mảng trên là: Chọn a/b/c

10000

32768

không xác định

|  |  |
| --- | --- |
| Kết quả: | b |

Kích thước của mẫu (tính theo bít hoặc byte)?

16 bits (2 bytes)

32 bits (4 bytes)

64 bits (8 bytes)

|  |  |
| --- | --- |
| Kết quả: | a |

Kích thước của toàn bản ghi (bộ đệm - biến mảng) tính theo KB?

16 KB or 16384

32 KB or 32768

64 KB, or 65536 bytes

|  |  |
| --- | --- |
| Kết quả: | c |

Địa chỉ phần tử đầu tiên trong mảng recBuff là:?

0x0103E8@DATA

0x010000@DATA

0x017F58@DATA

|  |  |
| --- | --- |
| Kết quả: | b |

Dữ liệu mảng recBuff được lưu trong bộ nhớ nào? Tham khảo file lnkx.cmd để có câu trả lời chính xác.

In the SARAM

In the DARAM

In the ROM

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Hiện đồ thị với 8000 mẫu đầu tiên của dữ liệu đã ghi. Trong menu Tools, chọn Graph, rồi chọn Single Time. Trong hộp thoại hiện ra, đặt các tham số như hình dưới rồi ấn OK.

Đặt Start Address là vị trí bắt đầu của mảng dữ liệu cần hiển thị trong bộ nhớ. Biến recBuff chứa địa chỉ này.

In đồ thị kết quả. Đồ thị dữ liệu trong mảng recBuff sẽ giống như mẫu trong hình sau.

Chú ý: Nếu không có máy in, có thể chụp màn hình rồi lưu lại trong tài liệu báo cáo thí nghiệm: Đồ thị dữ liệu mảng recBuff:

|  |
| --- |
|  |

* + - * 1. Điều chỉnh chuyển mạch DIP trong khối I/O về giá trị 1 để kích hoạt luật nén 8-bit μ-law.

Ấn nút  để chạy chương trình trên board.

* + - * 1. Ghi âm tín hiệu âm thanh (ấn nút INT0). Khoảng thời gian dài nhất ta có thể ghi được là?. Chọn a/b/c

Sấp xỉ 2 giây

Sấp xỉ 4 giây

Sấp xỉ 8 giây

|  |  |
| --- | --- |
| Kết quả: | c |

* + - * 1. Phát lại tính hiệu âm thanh đã ghi (Ấn nút INT1). Đânh giá chất lượng âm thanh? Chọn a/b/c.

Tốt hơn ghi 16 bit

Kém hơn ghi 16 bit một chút

Rất tệ

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Luật nén μ-law nhằm tối ưu giảm nhiễu lượng tử trong quá trình số hóa tín hiệu, đặc biệt là với tín hiệu nhỏ.

Điều chỉnh chuyển mạch DIP của khối I/O về giá trị 2 để kích hoạt chế độ lấy mẫu 8-bit không sử dụng luật nén μ-law và ấn nút INT1 để phát âm thanh đã ghi trong chế độ mới. Chất lượng âm thanh thế nào?. Chọn a/b/c

Tốt hơn ghi 16 bit

Kém hơn ghi 16 bit một chút

Rất tệ

|  |  |
| --- | --- |
| Kết quả: | c |

* + - * 1. Kích vào biểu tượng Icon

           Description automatically generated để dừng chương trình. Tín hiệu vẫn được lấy mẫu và lưu trong mảng recBuff. Tuy nhiên, mỗi phần tử của mảng (16 bits = 2 bytes) sẽ chứa 2 mẫu (1 byte / mẫu). Do đó, số mẫu lấy được sẽ gấp đôi so với trường hợp 16 bit

Tổng kích thước của bản ghi âm ở chế độ 8-bit là bao nhiêu?. Chọn a/b/c

16 KB or 16384

32 KB or 32768

64 KB, or 65536 bytes

|  |  |
| --- | --- |
| Kết quả: | c |

Dung lượng trên có tương ứng với thời gian ghi được trong chế độ này không?. Chọn a/b

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Điều chỉnh chuyển mạch DIP trong khối I/O đến giá trị 2 để kích hoạt chế độ 8-bit mode. Chế độ này không tối ưu nhiễu lượng tử khi số hóa tín hiệu. Ấn biểu tượng  để chạy chương trình.
        2. Ghi âm tín hiệu Audio đầu vào (ấn nút INT0). Khoảng thời gian dài nhất có thể ghi được là:? Chọn a/b/c

Sấp xỉ 2 giây

Sấp xỉ 4 giây

Sấp xỉ 8 giây

|  |  |
| --- | --- |
| Kết quả: | c |

* + - * 1. Phát lại tính hiệu âm thanh đã ghi (ấn nút INT1). Đánh giá chất lượng âm thanh? Chọn a/b/c

Kém hơn ghi 8-bit với luật nén μ-law một chút.

Tốt hơn ghi 16 bit.

Rất kém

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Nếu phát âm thanh được ghi trong chế độ 8-bit bằng chế độ 16-bit thì âm thanh phát ra sẽ như thế nào? Điều chỉnh chuyển mạch DIP của khối I/O về giá trị 0 để kích hoạt chế độ 16-bit.

Ấn nút INT1 để phát âm thanh trong chế độ mới. Âm thanh phát ra như thế nào?. Chọn a/b

Nghe rõ

Bị méo

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Lưu ý rằng dữ liệu được ghi trong cùng bảng (tại cùng địa chỉ như trước). Có thể dừng chương trình để quan sát đồ thị tín hiệu Audio đã ghi (xác định dữ liệu là dạng 8-bit long).
        2. Điều chỉnh chuyển mạch DIP của khối I/O đến giá trị 4 để kích hoạt chế độ xóa bộ nhớ flash.

Đảm bảo chương trình đang chạy (Nếu cần thiết, ấn lại nút  để chạy chương trình). Ấn phím INT0 để xóa dữ liệu đã ghi.

* + - * 1. Điều chỉnh chuyển mạch DIP trong khối I/O đến giá trị 3 để kích hoạt chế độ lấy mẫu 16-bit lưu trong bộ nhớ Flash.
        2. Tính thời gian dài nhất tín hiệu có thể ghi âm được trong bộ nhớ flash memory, biết rằng nó có dung lượng 8 MB (or 8192 KB).

Sử dụng thông tin trong file như ở bước 9 và 11 cho chế độ 16-bit mode. Chon a/b/c

8.5 giây.

8.5 phút

8.5 giờ

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Ghi âm tín hiệu Audio (ấn nút INT0).
        2. Phát lại tín hiệu Audio đã ghi (ấn nút INT1). Mô tả chất lương âm thanh ghi được ? Chọn a/b/c

tốt hơn mode 0

bằng mode 0

rất kém

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Kích vào biểu tượng Icon

           Description automatically generated để dừng chương trình. Bộ nhớ Flash kết nối bên ngoài DSP và được truy xuất từ các mạch ngoại vi. Vì thế không có cách đơn giản nào để quan sát nội dung của nó.
        2. Đến menu View trong phần mềm Code Composer và chọn Memory Browser.

Tìm địa chỉ 0x010000 trong cửa sổ Memory window. Xác định biến số tìm thấy trong phần DATA và kiểu giá trị của nó là 16-bit cơ số 16. Ghi lại nội dung của biến số?

|  |  |
| --- | --- |
| Kết quả: Giá trị của dữ liệu = | FFFF |

* + - * 1. Như được giải thích trong phần Cơ sở lý thuyết của bài thí nghiệm này, nếu cũng nội dung bộ nhớ được truy xuất bởi program bus (bus chương trình),nó cần được điều chỉnh địa chỉ để truy xuất theo byte thay cho theo words (4 byte).

Tìm đến địa chỉ 0x020000 trong cửa sổ Memory window. Xác định biến số tìm thấy trong phần PROGRAM kiểu giá trị của nó là 8-bit cơ số 16.

Nội dung của biến số theo byte?

|  |  |
| --- | --- |
| Kết quả: Giá trị của dữ liệu = | B0 |

Nội dung của dữ liệu trong byte kế tiếp (address 0x020001)?

|  |  |
| --- | --- |
| Kết quả: Giá trị của dữ liệu = | 6B |

* + - * 1. Click vào biểu tượng Shape, rectangle

           Description automatically generated để kết thúc chương trình. Quay lại tab Projects và thoát khỏi Code Composer.

Tắt nguồn điện và tháo các kết nối (dây dẫn) trên bảng mạch.

* + 1. Ex 4: Định địa chỉ.

Thiết lập kết nối trên bo mạch:

* + - * 1. Thực hiện các kết nối như trong hình. Có thể kết nối đầu vào của bộ tiền khuếch đại micrô với giắc cắm đầu ra âm thanh của máy tính nếu muốn sử dụng tín hiệu âm thanh từ máy tính của mình, hoặc sử dụng Microphone.
        2. Kết nối Bo mạch DSP với máy tính bằng cổng USB số 2. Đảm bảo rằng nguồn điện của Đế mạch đã được Bật và phần mềm Code Composer trên máy tính được cài đặt như giải thích trong Phụ Lục C. Chạy phần mềm Code Composer (CCS) trên máy tính.
        3. Kích hoạt dự án exercise\_2\_4 bằng cách nhấp chuột vào Project đó trong cửa sổ Project Explorer

Mở đồng thời hai file main.c và LVex2\_4.asm.

Chạy Debug bằng cách click vào biểu tượng A picture containing text, microscope, engine, device

Description automatically generated để tải chương trình vào chip DSP.

* + - * 1. Nhấn vào biểu tượng  để chạy chương trình. Kiểm tra thông tin hiển thị trên màn hình LCD của bo mạch DSP:

LabVolt 91031 Ex2-4

Signal Averaging.

Chương trình này cho phép DSP ghi tín hiệu tương tự vào bộ đệm xoay vòng chứa mười sáu mẫu. Giá trị trung bình của các mẫu được tính toán và kết quả được gửi ra dưới dạng đầu ra tương tự của DSP.

* + - * 1. Nói vào micrô (hoặc phát tệp âm thanh trên máy tính). Nghe âm thanh thu được từ DSP bằng loa trên Board hoặc sử dụng tai nghe.

Tín hiệu đầu ra bị ảnh hưởng bởi thuật toán DSP như thế nào? Chọn a/b/c/d

Âm thanh không sắc nét như tín hiệu ban đầu.

Tần số cao bị suy giảm.

Việc lấy trung bình có tác dụng một loại bộ lọc thông thấp.

Tất cả những điều trên

|  |  |
| --- | --- |
| Kết quả: | d |

* + - * 1. Kích vào nút Icon

           Description automatically generated để tạm dừng chương trình. Trong file LVex2\_4.asm, quan sát các lệnh trong phần \_initCB. Code trong phần này có tác dụng khởi tạo bộ đệm xoay vòng (circular buffer).

Số phần tử bộ đệm được đặt là 16, DSP được cấu hình để hoạt động với bộ đệm xoay vòng và chỉ số phần tử được khởi đầu là 0.

* + - * 1. Chuyển đến phần \_initDone của file LVex2\_4.asm và tìm lệnh mov mmap(@BK03), AC0.

Thêm điểm ngừng vào dòng này (dưới nhãn ;Breakpoint #01) và ấn nút  để chương trình chạy đến điểm ngừng đã đặt.

Loại địa chỉ nào được sử dụng cho biến BK03? Chọn a/b/c

Địa chỉ gián tiếp.

Địa chỉ trực tiếp.

Địa chỉ mặc định.

|  |  |
| --- | --- |
| Kết quả: | b |

Trong cửa sổ Disassembly, so sánh opcode với từng mã lệnh tương ứng. Trong các dòng sau, dòng nào là mã lệnh? Chọn a/b/c

a032\_98

a064\_98

c096\_98

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Đi tới lệnh amov #000000h, XAR1. Chèn một điểm ngừng tại dòng này (ngay bên dưới nhãn ;Breakpoint #02) và nhấn biểu tượng  để thực thi chương trình cho đến điểm đó.

Loại địa chỉ nào được sử dụng?. Chọn a/b/c

Đặt địa chỉ trực tiếp.

Đặt địa chỉ gián tiếp.

Đặt địa chỉ tức thời.

|  |  |
| --- | --- |
| Kết quả: | c |

Trong cửa sổ Disassembly, hãy quan sát mã thực thi (opcode) tương ứng với lệnh. Xác định đúng mã thực thi trong các dòng sau?. Chọn a/b/c

a931001977

ec319e000000

ea310c00194b

|  |  |
| --- | --- |
| Kết quả: | b |

Hằng số 000000 được sủ dụng như thế nào?. Chọn a/b/c

Hằng số (000000) được chứa trực tiếp trong opcode.

Hằng số (000000) đã được thay thế bằng một giá trị khác.

Hằng số (000000) không liên kết với mã máy.

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Đi tới lệnh mov AC1,\*AR1+. Chèn một điểm ngừng tại dòng này (ngay sau nhãn ;Breakpoint #03) và nhấn biểu tượng  để thực thi chương trình cho đến điểm đó.

Loại địa chỉ nào được sử dụng?. Chọn a/b/c

Địa chỉ trực tiếp.

Địa chỉ gián tiếp.

Địa chỉ tức thời.

|  |  |
| --- | --- |
| Kết quả: | b |

Trong cửa sổ Disassembly, quan sát mã thực thi (opcode) tương ứng với lệnh. Xác định mã thực thi trong các dòng sau?. Chọn a/b/c

c123

3c00

4c0f

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Thêm hai biến vào cửa sổ Expressions: biến average và recBuff. Biến average chứa tổng các mẫu chia cho 16, và đó là giá trị trung bình của các mẫu.

Biến recBuff là mảng chứa 16 phần tử trong bộ đệm vòng.

* + - * 1. Thay đổi giá trị các phần tử khác nhau của mảng recBuff từ cửa sổ Expressions. Số lượng mẫu lưu được trong mảng là bao nhiêu?. Chọn a/b/c

15

16

193A

|  |  |
| --- | --- |
| Kết quả: | b |

Kích thước (tính bằng bit) của mỗi mẫu là bao nhiêu?. Chọn a/b/c

32-bits

8-bits

16-bits

|  |  |
| --- | --- |
| Kết quả: | c |

* + - * 1. Đi đến lệnh HI(AC0<<#12),\*(\_average). Chèn một điểm ngừng tại dòng này (ngay sau nhãn ;Breakpoint #04) và nhấn biểu tượng  để thực thi chương trình cho đến điểm đó. Kết quả của việc lấy trung bình cộng các mẫu là:?. Chọn a/b/c

Giá trị trung bình của các mẫu được nhập trong mảng recBuff, được làm tròn xuống.

Tổng các mẫu được nhập trong mảng recBuff.

Tổng các mẫu được nhập trong bảng recBuff chia cho 15.

|  |  |
| --- | --- |
| Kết quả: | a |

Giá trị này có đúng là trung bình của các số đã nhập trong recBuff không? Chọn a/b

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Xóa tất cả các điểm dừng đã chèn cho đến nay.
        2. Thiết lập lại các kết nối như hình hướng dẫn.

Lưu ý: Mạch CODEC được thiết kế với điện áp đầu vào thấp, trong khoảng cực đại là ±0.9 V. Vì vậy khi cấp tín hiệu ngoài đến khối ANALOG INPUT, cần đảm bảo điện áp không vượt quá khoảng trên, nếu không mạch điện có thể hư hỏng không thể phục hồi.

* + - * 1. Dùng máy phát tín hiệu tạo ra dạng sóng tam giác có biên độ 0,4 V và tần số 100 Hz. Quan sát đồng thời tín hiệu đầu vào (Kênh 1) và tín hiệu đầu ra DSP (Kênh 2) trên máy hiện sóng.

Tín hiệu trông như thế nào?. Chọn a/b/c

Tín hiệu là dạng sóng vuông.

Tín hiệu là dạng sóng hình sin.

Tín hiệu là dạng sóng răng cưa.

|  |  |
| --- | --- |
| Kết quả: | b |

Tín hiệu đầu ra có đồng bộ với tín hiệu đầu vào không?. Chọn a/b

Có, hai tín hiệu đồng bộ.

Không, tín hiệu ra hơi trễ so với tín hiệu vào.

|  |  |
| --- | --- |
| Kết quả: | b |

Biên độ của tín hiệu đầu ra như thế nào?. Chọn a/b/c

Biên độ khoảng 1,1 V.

Biên độ khoảng 11,0 V.

Biên độ khoảng 0,11 V.

|  |  |
| --- | --- |
| Kết quả: | a |

Chất lượng của tín hiệu đầu ra như thế nào?. Chọn a/b/c

Tín hiệu đầu ra hoàn toàn đồng nhất với tín hiệu vào.

Tín hiệu đầu ra mờ hơn một chút so với tín hiệu ban đầu.

Hai tín hiệu hoàn toàn giống nhau.

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Tăng tần số của tín hiệu lên 800 Hz. Quan sát đồng thời tín hiệu đầu vào (Kênh 1) và tín hiệu đầu ra DSP (Kênh 2) trên máy hiện sóng.

Tín hiệu đầu ra có đồng bộ với tín hiệu đầu vào không?. Chọn a/b/c

Có, hai tín hiệu đồng bộ.

Không, tín hiệu ra hơi trễ so với tín hiệu vào.

|  |  |
| --- | --- |
| Kết quả: | b |

Biên độ của tín hiệu đầu ra như thế nào?. Chọn a/b/c

Biên độ khoảng 1,1 V.

Biên độ khoảng 0,25 V.

Biên độ khoảng 4,0 V.

|  |  |
| --- | --- |
| Kết quả: | c |

Chất lượng của tín hiệu đầu ra như thế nào?. Chọn a/b/c

Tín hiệu đầu ra mờ hơn tín hiệu đầu vào.

Hai tín hiệu hoàn toàn giống nhau.

Tín hiệu đầu ra sạch hơn nhiều so với tín hiệu đầu vào.

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Thay đổi tần số của tín hiệu về 450 Hz. Quan sát đồng thời tín hiệu đầu vào (Kênh 1) và tín hiệu đầu ra DSP (Kênh 2) trên máy hiện sóng.

Tín hiệu đầu ra có đồng bộ với tín hiệu đầu vào không?. Chọn a/b/c

Có, hai tín hiệu đồng bộ.

Không, tín hiệu ra hơi trễ so với tín hiệu vào.

|  |  |
| --- | --- |
| Kết quả: | a |

Biên độ của tín hiệu đầu ra như thế nào?. Chọn a/b/c

Biên độ khoảng 0.125 V.

Biên độ khoảng 1,25 V.

Biên độ khoảng 12,5 V.

|  |  |
| --- | --- |
| Kết quả: | a |

Chất lượng của tín hiệu đầu ra như thế nào?. Chọn a/b/c

Tín hiệu đầu ra mờ hơn tín hiệu đầu vào.

Hai tín hiệu hoàn toàn giống nhau.

Tín hiệu đầu ra sạch hơn nhiều so với tín hiệu đầu vào.

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Điều chỉnh máy phát tạo ra tín hiệu dạng sóng tam giác có biên độ 0,4 V và tần số 100 Hz. Quan sát đồng thời tín hiệu đầu vào (Kênh 1) và tín hiệu đầu ra DSP (Kênh 2) trên máy hiện sóng.

Tín hiệu đầu ra như thế nào?. Chọn a/b/c/d

Tín hiệu gần giống với dạng sóng hình sin.

Thuật toán trung bình làm mịn tín hiệu gốc.

Có một khoảng trễ giữa hai tín hiệu.

Tất cả những điều trên.

|  |  |
| --- | --- |
| Kết quả: | d |

* + - * 1. Điều chỉnh máy phát tạo ra tin hiệu sóng vuông có biên độ 0,4 V và tần số 180 Hz. Quan sát cả tín hiệu đầu vào (Kênh 1) và tín hiệu đầu ra DSP (Kênh 2) trên máy hiện sóng.

Tín hiệu đầu ra như thế nào?. Chọn a/b/c/d

Tín hiệu giống như một dạng sóng tam giác.

Tín hiệu bão hòa (cắt bớt ở đỉnh tam giác), vì giá trị trung bình không thể cao hơn giá trị cực đại.

Có một khoảng trễ giữa hai tín hiệu.

Tất cả những điều trên.

|  |  |
| --- | --- |
| Kết quả: | d |

* + - * 1. Click vào biểu tượng Shape, rectangle

           Description automatically generated để kết thúc chương trình. Quay lại tab Projects và thoát khỏi Code Composer.

Tắt nguồn điện và tháo các kết nối (dây dẫn) trên bảng mạch.

* 1. U3: Vào/Ra và các Ngoại vi.
     1. Ex 1: Ứng dụng với I/O và Ngoại vi.

Thiết lập kết nối trên bo mạch thí nghiệm

* + - * 1. Thực hiện kết nối như hình dưới. Có thể kết nối đầu vào của bộ tiền khuếch đại micrô với giắc cắm đầu ra âm thanh của máy tính.
        2. Kết nối Bo mạch DSP với máy tính bằng cổng USB số 2. Đảm bảo rằng nguồn điện của Đế mạch đã được Bật và phần mềm Code Composer trên máy tính được cài đặt như giải thích trong Phụ Lục C. Chạy phần mềm Code Composer (CCS) trên máy tính.
        3. Kích hoạt dự án exercise\_3\_1 bằng cách nhấp chuột vào Project đó trong cửa sổ Project Explorer. Mở file main.c của bài tập này. Chạy Debug bằng cách click vào biểu tượng A picture containing text, microscope, engine, device

           Description automatically generated để tải chương trình vào chip DSP.
        4. Quan sát file main.c. Hàm khởi tạo nào sau đây được gọi trong chương trình chính?. Chọn a/b/c

interrupt\_init();// Khởi tạo ngắt

EZDSP5535\_init();// Kích hoạt đồng hồ cho tất cả các thiết bị ngoại vi

initDisplay();

Iaic3204\_init();// Khởi tạo CODEC

|  |  |
| --- | --- |
| Kết quả: |  |

* + - * 1. Xác định vị trí lệnh gọi hàm I2S\_initForDMA() trong tệp main.c. Di con trỏ chuột qua cuộc gọi và thực hiện lệnh Open Declaration (Ctrl + F3 hoặc nhấp chuột phải và chọn lệnh từ menu).

Quan sát rằng hàm này tạo ra một biến, hwConfig, thuộc loại I2S\_Config. Biến này là một cấu trúc được tạo thành từ một số biến khác. Nhiều trong số này được khởi tạo trong hàm I2S\_initForDMA. Ví dụ: biến dataType được đặt để bật bản ghi âm thanh nổi.

Quay lại chức năng chính.

* + - * 1. Xác định vị trí lệnh gọi hàm start\_acq() trong hàm chính. Di con trỏ chuột qua cuộc gọi và thực hiện lệnh Open Declaration (Ctrl + F3 hoặc nhấp chuột phải và chọn lệnh từ menu).

Trình soạn thảo mã sẽ đưa bạn đến mã của hàm start\_acq().

Trong nhiều dòng lệnh, một số lượng lớn tham số được đặt cho kênh DMA 4 và 5. Tìm dòng lệnh sau cho kênh 5 (I2S write):

dmaConfig.destAddr = (Uint32)I2S2TXRT0;

Dòng này chỉ định địa chỉ nơi dữ liệu từ CODEC sẽ được ghi vào bộ nhớ DSP khi nó được truyền.

Di chuột qua định nghĩa I2S2TXRT0 và thực hiện lệnh Open Definition. Tác vụ này sẽ đưa đến vị trí trong mã nơi địa chỉ này được xác định và cho thấy địa chỉ đích là 0x2A0C.

Các Thanh ghi Ngoại vi

* + - * 1. Dừng chương trình bằng nút Icon

           Description automatically generated. Mở cửa sổ Registers và tìm các thanh ghi khác nhau liên quan đến thiết bị ngoại vi.

Ta sẽ thấy một danh sách dài, bắt đầu bằng DMA0 và tiếp tục cho các thiết bị ngoại vi khác nhau. Khám phá các menu phụ để xem phạm vi tham số có sẵn cho các thiết bị ngoại vi.

* + - * 1. Quan sát menu thanh ghi TIMER0. Quan sát thấy có bit kích hoạt để kích hoạt bộ hẹn giờ. (TIMER0 ▶ TCR ▶ TIMEN).

Bật bộ hẹn giờ bằng cách đặt bit này thành 1.

Đặt thanh ghi PSCDIV thành 0x9.

Bắt đầu Kích hoạt TIMER bằng cách đặt bit START thành 1.

Nhấn mũi tên  vài lần và quan sát kết quả trên thanh ghi TIMCNT1 và TIMCNT2. Giá trị TIMCNT2</look> sẽ giảm đều đặn sau mỗi lần nhấp vào mũi tên  vì bộ đếm thời gian giảm giá trị.</p>

* + - * 1. Xác định vị trí menu thanh ghi I2S2 và mở nó. Mở menu I2SSCTRL và tìm bit ENABLE. Bit này phải được đặt thành 1.

Xóa bit ENABLE thành 0.

Thao tác này sẽ vô hiệu hóa thiết bị ngoại vi I2S được sử dụng trong ứng dụng này.

* + - * 1. Nhấn vào biểu tượng  để chạy chương trình. Nhấn nút INT1 để phát lại tín hiệu đã ghi. Ta nhận thấy gì?. Chọn a/b

Không phát lại âm thanh

Phát lại âm thanh

|  |  |
| --- | --- |
| Kết quả: |  |

* + - * 1. Nhấn nút INT0 để kích hoạt chế độ ghi và ghi tín hiệu mới. Dừng chương trình bằng nút Icon

           Description automatically generated và đặt bit ENABLE lên 1 (I2S2 ► I2SSCTRL ► ENABLE) để kích hoạt lại khối ngoại vi I2S.
        2. Nhấn nút chạy  để chạy chương trình. Nhấn nút INT1 để phát lại tín hiệu đã ghi. Ta nhận được gì?. Chọn a/b

Phát lại âm thanh mới được ghi.

Phát lại âm thanh đã ghi trước đó.

|  |  |
| --- | --- |
| Kết quả: |  |

* + - * 1. Dừng chương trình bằng nút Icon

           Description automatically generated.

Trong cửa sổ Registers, chỉnh sửa tham số DMA1 ▶ DMACH1DSAL thành 0x2A0D. Điều này thay đổi địa chỉ đích cho dữ liệu truyền qua kênh DMA 5 tới CODEC.

* + - * 1. Nhấn nút chạy  để chạy chương trình. Nhấn nút INT1 để phát lại tín hiệu đã ghi. Ta nhận thấy?. Chọn a/b

Phát lại âm thanh.

Không phát lại âm thanh.

|  |  |
| --- | --- |
| Kết quả: |  |

Nhấn nút INT1 để phát lại tín hiệu đã ghi.

* + - * 1. Dừng chương trình bằng nút Icon

           Description automatically generated.

Trong cửa sổ Registers, chỉnh sửa tham số DMA1 ▶ DMACH1DSAL thành 0x2A0C. Điều này sửa địa chỉ đích để truyền dữ liệu qua kênh DMA 5 tới CODEC.

* + - * 1. Nhấn nút chạy  để chạy chương trình. Nhấn nút INT1 để phát lại tín hiệu đã ghi. Ta nhận thấy?. Chọn a/b

Phát lại âm thanh

Không phát lại âm thanh

|  |  |
| --- | --- |
| Kết quả: |  |

Việc thay đổi địa chỉ đích của kênh DMA 5 khiến việc truyền dữ liệu từ DSP sang CODEC không thể thực hiện được nhưng dữ liệu vẫn có thể đi từ CODEC sang DSP qua kênh 4.

* + - * 1. Click vào biểu tượng Shape, rectangle

           Description automatically generated để kết thúc chương trình. Quay lại tab Projects và thoát khỏi Code Composer.

Tắt nguồn điện và tháo các kết nối (dây dẫn) trên bảng mạch.

* 1. U4: Xử lý thời gian thực.
     1. Ex 1: Lấy mẫu và chuyển đổi ADC/DAC.

Ứng dụng Lấy mẫu tín hiệu đầu vào

Một tín hiệu tương tự với có biên độ tối đa khoảng ±0,75 Vpp được cấp đến ĐẦU VÀO ANALOG. Tín hiệu này được lọc bằng bộ lọc chống chồng phổ.

Chú ý: Điện áp vào lớn hơn ±0,8 Vpp có thể dẫn đến hiện tượng méo do cắt ngưỡng.

CODEC liên tục lấy mẫu ĐẦU VÀO ANALOG đã lọc ở tốc độ lấy mẫu được xác định bởi công tắc INPUT DIP và thực hiện chuyển đổi A/D để thu được dữ liệu số. (Tốc độ lấy mẫu thay đổi ngay lập tức khi bạn thay đổi cài đặt công tắc DIP; không cần thiết phải nhấn nút ngắt).

CODEC ngay lập tức thực hiện chuyển đổi D/A của dữ liệu số. Điện áp đầu ra được giữ không đổi giữa các mẫu. Bộ lọc sau lọc tín hiệu đã chuyển đổi D/A nhằm cố gắng tái tạo lại tín hiệu tương tự bằng cách lọc ra các bản sao.

Có hai chế độ đầu ra: Dạng sóng thời gian và Phổ biên độ:

* Chế độ dạng sóng: Tín hiệu tương tự được tái tạo có sẵn ở ĐẦU RA ANALOG. Dạng sóng này có thể được quan sát trên máy hiện sóng.
* Chế độ Phổ biên độ: Phổ tần số của tín hiệu tương tự được tái tạo có sẵn tại ANALOG OUTPUT. Phổ tần số này có thể được quan sát trên máy hiện sóng.

Chế độ đầu ra được chọn bằng nút INT0. Chế độ đầu ra đã chọn được hiển thị trên màn hình LCD.

Bộ lọc thông thấp hạn chế băng thông ở đầu vào của CODEC và bộ lọc sau ở đầu ra của CODEC có thể được bật hoặc tắt cùng nhau bằng cách sử dụng nút INT1.

Trạng thái của bộ lọc được hiển thị trên màn hình LCD. Cả hai bộ lọc này đều là bộ lọc thông thấp. Tần số cắt 3 dB của các bộ lọc này được cố định ở khoảng 3100 Hz.

Màn hình LCD hiển thị tốc độ lấy mẫu, chế độ đầu ra và trạng thái của bộ lọc.

* + - * 1. Kết nối bo mạch DSP với máy tính của bạn bằng cổng USB số 2 trên bo mạch và cổng USB tiêu chuẩn trên máy tính của bạn.

Đảm bảo rằng nguồn điện của bảng mạch đã được BẬT và phần mềm Code Composer được cài đặt như được giải thích trong Phụ lục C.

Đặt công tắc INPUT DIP thành 0000 0000. Khởi chạy Trình soạn thảo mã CCS trên máy tính của bạn.

* + - * 1. Chuyển Exercise\_4\_1 thành dự án đang hoạt động bằng cách nhấp chuột vào dự án đó trong cửa sổ Project Explorer.

Mở tệp interrupt.c của bài tập này.

Nếu cửa sổ Breakpoints không hiển thị, hãy chọn Breakpoints</i></look> trong menu View. Nếu có bất kỳ điểm dừng nào được hiển thị trong cửa sổ này, hãy nhấp chuột phải vào cửa sổ đó và chọn Remove All.

Khởi chạy chế độ Gỡ lỗi bằng cách nhấp vào biểu tượng A picture containing text, microscope, engine, device

Description automatically generated để tải chương trình vào DSP.

* + - * 1. Nhấn nút chạy  để chạy chương trình trên bo mạch thí nghiệm. Tại thời điểm này, ta sẽ thấy thông tin trên màn hình LCD của bảng mạch DSP như sau:

LabVolt 91031 Ex4-1

Sampling

Rate = 12000,0 Hz

Waveform, Filters OFF

* + - * 1. Thực hiện các kết nối như được hiển thị. Đầu ra của bộ tạo tín hiệu phải được kết nối với Ch 1 của máy hiện sóng. ĐẦU RA ANALOG của CODEC phải được kết nối với Ch 2.

Trước khi kết nối bộ tạo tín hiệu, hãy điều chỉnh mức đầu ra của nó thành 0 V.

Chú ý: CODEC được thiết kế cho tín hiệu điện áp thấp có điện áp đầu vào tối đa ±0,9 V. Khi sử dụng nguồn tín hiệu bên ngoài, hãy đảm bảo điện áp ở ĐẦU VÀO ANALOG nằm trong phạm vi này. Vượt quá phạm vi điện áp này có thể làm hỏng bo mạch.

* + - * 1. Bảng này hiển thị cài đặt công tắc DIP để sử dụng cho các tốc độ lấy mẫu khác nhau. Ứng dụng này có tốc độ lấy mẫu nội bộ hiệu quả là 12 000 Hz (12 000 mẫu mỗi giây). Tỷ lệ lấy mẫu thấp hơn có được bằng cách chia mẫu theo hệ số thay đổi từ 1 đến 9.

Chú ý: Tất cả tốc độ lấy mẫu có sẵn được hiển thị trong bảng. Sử dụng cài đặt công tắc DIP khác với cài đặt được hiển thị sẽ dẫn đến tốc độ lấy mẫu tối thiểu.

Màn hình LCD hiển thị tốc độ lấy mẫu. Nó cũng hiển thị chế độ đầu ra (Dạng sóng hoặc Quang phổ) cũng như trạng thái của bộ lọc chống chồng phổ và bộ lọc đầu ra DAC (TẮT hoặc BẬT).

Sử dụng công tắc DIP để đặt tốc độ lấy mẫu thành 12 000 Hz (DIP SW: 0000 0001).

Sử dụng nút INT0 để đặt chế độ đầu ra thành Dạng sóng. Sử dụng nút INT1 để TẮT bộ lọc.

Chú ý: Có thể phải nhấn nút ngắt nhiều lần để có được cài đặt mong muốn.

Điều chỉnh bộ tạo tín hiệu để tạo ra tín hiệu sóng hình sin có biên độ 0,7 Vpp. Tần số của sóng hình sin là 300 Hz.

Tín hiệu trên máy hiện sóng có dạng như hình minh hoạ. Tín hiệu đầu ra so với tín hiệu đầu vào như thế nào?. Chọn a/b/c

Tín hiệu đầu ra là dạng sóng từng bước có tần số gấp đôi tần số tín hiệu đầu vào.

Tín hiệu đầu ra là dạng sóng có bước sóng với tần số bằng một nửa tần số tín hiệu đầu vào.

Tín hiệu đầu ra là dạng sóng từng bước có cùng hình dạng và tần số với tín hiệu đầu vào.

|  |  |
| --- | --- |
| Kết quả: | c |

* + - * 1. Sử dụng con trỏ trên máy hiện sóng để xác định khoảng thời gian lấy mẫu, tức là khoảng thời gian của mỗi bước trong dạng sóng đầu ra. Sau đó tính tốc độ lấy mẫu.

Đo thời lượng của một số bước, như minh họa trong Hình (hiển thị 5 bước) và tìm thời lượng trung bình của bước. Xác định giá trị và ghi vào bảng sau:

|  |  |
| --- | --- |
| Thời gian lấy mẫu gần đúng (ms) = | 0.084 |
| Tốc độ lấy mẫu gần đúng (Hz) = | 12000 |

* + - * 1. Trong Trình soạn thảo CCS, tạm dừng chương trình bằng nút Icon

           Description automatically generated. Trong tệp interrupt.c, tìm dòng chú thích //Breakpoint #01.

Đặt điểm ngắt trên dòng ngay bên dưới nhận xét này bằng cách nhấp đúp vào bên trái số dòng.

Nhấn nút  để chạy chương trình. Nó sẽ dừng sau khi lấy mẫu từ tín hiệu ANALOG INPUT, thông qua bộ đệm DMA và đặt dữ liệu này vào vectơ x.

Có thể hiển thị đồ thị của vectơ x trong Trình soạn thảo mã. Trong trình đơn Tools, chọn Graph, sau đó chọn Single Time.

Thiết lập các thông số như trong hình.

Hình này hiển thị quá trình thực thi bị dừng tại điểm dừng và hiển thị biểu đồ ví dụ về các giá trị trong x. (Điểm bắt đầu của dạng sóng sẽ khác nhau mỗi lần thực thi đạt đến điểm dừng này).

Biểu đồ của các giá trị trong x có giống với dạng sóng tín hiệu đầu ra được hiển thị trên máy hiện sóng không?. Chọn a/b

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Trong Code Composer, hãy tắt hoặc xóa điểm ngắt. Nhấn nút chạy  Sử dụng nút INT1 để BẬT bộ lọc.

Luy ý: Như đã đề cập trong Ứng dụng lấy mẫu được sử dụng trong bài tập này, tần số cắt 3 dB của các bộ lọc này được cố định ở khoảng tần số 3100 Hz.

Bộ lọc chống chồng phổ có tác dụng gì trong điều kiện hiện tại?. Chọn a/b

Nó ít ảnh hưởng đến dạng sóng đầu vào vì tần số cắt (3100 Hz) của nó cao hơn nhiều so với tần số tín hiệu đầu vào (300 Hz).

Nó có ảnh hưởng lớn đến dạng sóng đầu vào vì tần số cắt (3100 Hz) của nó cao hơn nhiều so với tần số tín hiệu đầu vào (300 Hz).

|  |  |
| --- | --- |
| Kết quả: | a |

Bộ lọc đầu ra có tác dụng gì với tín hiệu đầu ra?. Chọn a/b

Không có tác dụng

Nó làm mượt các bước của dạng sóng đầu ra.

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Tăng từ từ tần số tín hiệu đầu vào lên khoảng 3000 Hz. Bạn có thể nghe tín hiệu đầu ra khi thực hiện thay đổi. Tín hiệu đầu ra có giống với tín hiệu analog ban đầu không?. Chọn a/b.

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Tăng từ từ tần số tín hiệu đầu vào lên khoảng 4000 Hz. Chuyện gì xảy ra?Chọn a/b/c

Vì các bộ lọc được BẬT nên biên độ tín hiệu đầu ra giảm nhanh khi tần số tín hiệu đầu vào tăng vượt quá 3000 Hz.

Ở 4000 Hz, biên độ tín hiệu đầu ra bằng 0.

Tất cả những điều trên.

|  |  |
| --- | --- |
| Kết quả: | c |

Xác định tốc độ Nyquist cho tốc độ lấy mẫu đã chọn. ƒNyquest= <ƒs/2.

|  |  |
| --- | --- |
| fs = | 12000 |
| ƒNyquest = | 6000 |

Mục đích của bộ lọc anti-aliasing filter là?. Chọn a/b/c/d

Nó loại bỏ các thành phần tần số cao khỏi tín hiệu đầu vào có thể dẫn đến hiện tượng răng cưa.

Vì tần số cắt 3 dB của bộ lọc là khoảng 3100 Hz nên bộ lọc sẽ loại bỏ hoàn toàn các tần số trên 4000 Hz.

4000Hz thấp hơn nhiều so với tốc độ Nyquist.

Tất cả những điều trên

|  |  |
| --- | --- |
| Kết quả: | d |

Mục đích của bộ lọc anti-aliasing filter là?. Chọn a/b/c/d

Nó loại bỏ các thành phần tần số cao khỏi tín hiệu đầu vào có thể dẫn đến hiện tượng chồng phổ.

Vì tần số cắt 3 dB của bộ lọc là khoảng 3100 Hz nên bộ lọc sẽ loại bỏ hoàn toàn các tần số trên 4000 Hz.

4000Hz thấp hơn nhiều so với tốc độ Nyquist.

Tất cả những điều trên.

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Đặt tốc độ lấy mẫu thành 6000 Hz (đặt công tắc DIP thành 0000 0010). Đối chiếu kết quả thể hiện theo hình mình hoạ. Mô tả tín hiệu đầu ra?. Chọn a/b/c

dạng sóng hình sin ở tần số cao hơn tín hiệu đầu vào.

dạng sóng hình sin ở tần số thấp hơn tín hiệu đầu vào.

dạng sóng hình sin có cùng tần số với tín hiệu đầu vào.

|  |  |
| --- | --- |
| Kết quả: | b |

Tần số của dạng sóng đầu ra là bao nhiêu?.Nếu cần, bạn có thể sử dụng con trỏ dọc để xác định tần số.Tần số của dạng sóng đầu ra là:

|  |  |
| --- | --- |
| Ƒout (Hz) = | 2000 |

Giải thích điều gì đã xảy ra bằng cách trả lời những câu sau:

Tốc độ Nyquist hiện là ƒs/2 = 6000/2 = 3000 Hz. Chọn a/b

|  |  |
| --- | --- |
| Kết quả: | a |

Vì tần số tín hiệu đầu vào lớn hơn tốc độ Nyquist nên hiện tượng chồng phổ đã xảy ra. Chọn a/b

Đúng

Sai

|  |  |
| --- | --- |
| Kết quả: | a |

Dạng sóng đầu ra là dạng sóng của bản sao đầu tiên. Tần số của nó là ƒs –ƒm = 6000 - 4000 = 2000 Hz. Chọn a/b

Đúng

Sai

|  |  |
| --- | --- |
| Kết quả: | a |

Chú ý: Vì tần số cắt của bộ lọc khử răng cưa là khoảng 3100 Hz nên bộ lọc này không thể ngăn chặn hiện tượng răng cưa trong những điều kiện này.

* + - * 1. Đặt tần số của tín hiệu đầu vào thành 1000 Hz. Sử dụng nút INT1 để TẮT bộ lọc.

Đặt tốc độ lấy mẫu cho từng giá trị trong Bảng trong khi quan sát dạng sóng đầu ra trên máy hiện sóng.

Mô tả dạng sóng đầu ra với tốc độ lấy mẫu được đặt thành 2000 Hz và giải thích điều gì đang xảy ra. Chọn a/b/c

Dạng sóng đầu ra trùng lặp chính xác với tín hiệu đầu vào. Bốn mẫu được lấy trong mỗi chu kỳ.

Dạng sóng đầu ra gần như là sóng vuông có cùng tần số với tín hiệu đầu vào. Bốn mẫu được lấy trong mỗi chu kỳ.

Dạng sóng đầu ra gần như là sóng vuông có cùng tần số với tín hiệu đầu vào. Hai mẫu được lấy trong mỗi chu kỳ.

|  |  |
| --- | --- |
| Kết quả: | c |

* + - * 1. Đặt tốc độ lấy mẫu thành 1714 Hz (DIP SW: 0000 0111) rồi đến 1500 Hz (DIP SW: 0000 1000). Tín hiệu đầu ra là dạng sóng bậc thang?. Chọn a.b

Đúng

Sai

|  |  |
| --- | --- |
| Kết quả: | a |

Tần số của dạng sóng đầu ra là bao nhiêu?

|  |  |
| --- | --- |
| Tần số của dạng sóng đầu ra là (Hz)= | 500 |

Giải thích điều gì đã xảy ra bằng cách trả lời những câu sau:

Tần số Nyquist hiện làƒs/2 = 1500⁄2 = 750 Hz. Chọn a/b

Đúng

Sai

|  |  |
| --- | --- |
| Kết quả: | a |

Vì tần số tín hiệu đầu vào ƒm lớn hơn tốc độ Nyquist nên hiện tượng chồng phổ KHÔNG xảy ra. Chọn a/b

Đúng

Sai

|  |  |
| --- | --- |
| Kết quả: | b |

Dạng sóng đầu ra là dạng sóng của biên tần thấp đầu tiên. Tần số của nó là ƒs – ƒm = 1500 - 1000 = 500 Hz. Chọn a/b

Đúng

Sai

|  |  |
| --- | --- |
| Kết quả: | a |

Quan sát trong miền Tần số.

Có thể thực hiện quan sát miền tần số bằng cách đặt ứng dụng ở chế độ Spectrum. Ở chế độ này, tín hiệu ở đầu ra tương tự của CODEC chứa phổ tần số có thể được hiển thị trên máy hiện sóng tiêu chuẩn. Điều này sẽ cho phép bạn thực hiện các quan sát định tính trong miền tần số.

Ở chế độ này, khi các bộ lọc BẬT, tín hiệu đầu ra sẽ được lọc bởi bộ lọc sau trước khi tính phổ. Điều này cho phép bạn hiển thị phổ của tín hiệu đã lọc.

Mặc dù bạn cũng có thể sử dụng chế độ Phổ để xác định tần số gần đúng của các thành phần tần số trong tín hiệu nhưng các phép đo tần số ở chế độ này rất không chính xác.

* + - * 1. Sử dụng công tắc DIP để đặt tốc độ lấy mẫu thành 12000 (DIP SW: 0000 0001). Sử dụng nút INT1 để đặt chế độ đầu ra thành Spectrum.

Sử dụng nút INT0 để TẮT các bộ lọc.

Điều chỉnh mức đầu ra của bộ tạo tín hiệu về 0V.

Trên máy hiện sóng: đặt Cơ sở thời gian thành 0,5 ms/div. đặt Tỷ lệ Kênh 2 thành 50 mV/div.

Đặt Nguồn kích hoạt thành Ch2, đặt Độ dốc kích hoạt thành Sườn âm và đặt Mức kích hoạt khoảng -0,05 V.

Máy hiện sóng sẽ kích hoạt ở đỉnh âm. Nếu cần, hãy di chuyển điểm kích hoạt theo chiều ngang để xem đỉnh này, như trong Hình minh hoạ.

Chú ý: Điều chỉnh Mức kích hoạt (triger trên máy hiện sóng), để tạo ra màn hình ổn định.

* + - * 1. Di chuyển vị trí kích hoạt theo chiều ngang sang cực bên trái của lưới. Điều chỉnh bộ tạo tín hiệu để tạo ra tín hiệu thông báo sóng hình sin có biên độ 0,7 Vpp.

Đặt tần số ƒm của sóng hình sin thành 1000 Hz. Bạn sẽ thấy một đỉnh dương ở gần giữa phần đầu tiên, như minh họa trong Hình.

Đỉnh dương này biểu thị tần số hình sin là 1000 Hz. Hiển thị con trỏ dọc và đặt con trỏ dọc ở giữa đỉnh dương này. Không di chuyển con trỏ này trong phần còn lại của bài tập này. Máy hiện sóng sẽ chỉ ra vị trí nằm ngang của con trỏ này tính bằng mili giây.

Cạnh cực bên trái của đồ thị Ch 2 tương ứng với 0 Hz (dc). Vị trí tính bằng mili giây của con trỏ bạn di chuyển qua đỉnh dương tương ứng với 1000 Hz.

* + - * 1. Đặt tần số ƒm của tín hiệu đầu vào thành 3000 Hz. Đỉnh dương nhất bên trái bây giờ sẽ nằm ở phần thứ hai của lưới và ba đỉnh khác sẽ hiển thị, như minh họa trong Hình này.

Định vị con trỏ dọc thứ hai theo chiều dọc ở giữa đỉnh dương ngoài cùng bên trái, như minh họa trong cùng Hình. Vị trí nằm ngang này tương ứng với 3000 Hz. (Không di chuyển con trỏ ở tốc độ 1000 Hz).

Nhập vị trí nằm ngang (tính bằng ms) tương ứng với 3000 Hz ở hàng đầu tiên của Bảng.

Sử dụng con trỏ thứ hai để xác định vị trí nằm ngang của ba đỉnh khác và nhập chúng vào Bảng.

Thang tần số của màn hình Spectrum là khoảng 4500 Hz/ms. Sử dụng thang đo này để xác định tần số gần đúng của ba đỉnh dương cuối cùng trong Bảng.

Chú ý: Làm tròn câu trả lời của bạn đến 5000 Hz gần nhất.

|  |  |  |  |
| --- | --- | --- | --- |
| Đỉnh từ bên trái | Vị trí ngang (ms) | Tần số Fpeak (Hz) | Tính toán tần số fpeak theo fm và fs |
| 1 | 0.7 | 3000 | Fpeak = fm = 3000Hz |
| 2 | 2.03 | 9000 | Fpeak = fs - fm |
| 3 | 3.35 | 15000 | Fpeak = fs + fm |
| 4 | 4.7 | 21000 | Fpeak = 2fs - fm |

* + - * 1. Định vị con trỏ thứ hai ở tần số 3000 Hz, như trong Hình. Con trỏ này và con trỏ ở 1000 Hz sẽ dùng làm tham chiếu tần số. Đặt tần số của sóng hình sin đầu vào thành tần số đầu tiên trong Bảng.

Đặt tốc độ lấy mẫu thành 6000 Hz (DIP SW: 0000 0010).

Tốc độ Nyquist cho tốc độ lấy mẫu hiện tại là bao nhiêu?. Chọn a/b/c

2000

3000

6000

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Trong khi quan sát phổ tần số trên máy hiện sóng, hãy đặt tần số của sóng hình sin thành các tần số khác trong Bảng.

Lưu ý: Bạn có thể tìm thấy các ảnh chụp màn hình máy hiện sóng ví dụ trong hướng dẫn dành cho giao viên về các giá trị khác nhau của ƒm được tìm thấy trong bảng.

Giải thích hiện tượng xảy ra khi tần số sóng hình sin đầu vào tiến gần đến tốc độ Nyquist: Khi tần số tín hiệu sóng hình sin tăng lên, đỉnh của nó trong màn hình quang phổ sẽ di chuyển sang bên phải. Đồng thời, các đỉnh sóng biên tần bậc nhất lan rộng ra. Chọn a/b

Đúng

Sai

|  |  |
| --- | --- |
| Kết quả: | a |

Giải thích hiện tượng xảy ra khi tần số sóng hình sin đầu vào đạt đến tốc độ Nyquist: Chọn a/b

Sóng hài đầu tiên thấp hơn nhiều so với tín hiệu do có hiện tượng chồng phổ

Tín hiệu và thành phần bên dưới của sóng hài đầu tiên trùng khớp với nhau.

|  |  |
| --- | --- |
| Kết quả: | b |

Giải thích điều gì xảy ra khi tần số sóng hình sin đầu vào vượt quá tốc độ Nyquist. Chọn a/b

Thành phần bên dưới biên tần thứ nhất chuyển xuống dưới tần số tín hiệu và xảy ra hiện tượng chồng phổ.

Tín hiệu và thành phần bên dưới biên tần thứ nhất trùng khớp với nhau.

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Khi tần số tín hiệu đầu vào được đặt thành 5000 Hz, hãy đặt Chế độ thành Dạng sóng.

Điều chỉnh lại tỷ lệ Kênh 2. Mô tả những gì bạn quan sát được?. Chọn a/b/c

Dạng sóng đầu ra là dạng sóng bước có tần số 1000 Hz.

Dạng sóng đầu ra là dạng sóng bước có tần số 2500 Hz.

Dạng sóng đầu ra là dạng sóng bước có tần số 5000 Hz.

|  |  |
| --- | --- |
| Kết quả: | a |

Sử dụng cấu hình hiện tại, BẬT bộ lọc và giải thích những gì bạn quan sát được.

Bộ lọc bài làm mịn dạng sóng đầu ra. Tuy nhiên, tần số của dạng sóng này cao hơn nhiều so với tần số của dạng sóng đầu vào do có hiện tượng chồng phổ. Chọn a/b

Đúng

Sai

|  |  |
| --- | --- |
| Kết quả: | b |

Tần số của dạng sóng đầu ra là tần số của thành phần thấp hơn của biên tần thứ nhất, tức là ƒs – ƒm = 6000 - 5000 = 1000 Hz. Chọn a/b

Đúng

Sai

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Đặt tần số của sóng hình sin đầu vào thành 2250 Hz. BẬT và TẮT bộ lọc. So sánh tín hiệu đầu vào và đầu ra khi bộ lọc BẬT. Bộ lọc có tái tạo lại tín hiệu đầu ra một cách chính xác không?. Chọn a/b

Có, tín hiệu đầu ra có cùng tần số với tín hiệu đầu vào.

Không, tín hiệu đầu ra có tần số khác với tín hiệu đầu vào

|  |  |
| --- | --- |
| Kết quả: | a |

Đặt chế độ thành hiển thị phổ. Trên máy hiện sóng, điều chỉnh Thang đo Kênh 2 thành 50 mV/div.

BẬT và TẮT bộ lọc trong khi quan sát phổ của tín hiệu đầu ra. Bộ lọc đầu ra tái tạo lại tín hiệu đầu ra như thế nào?. Trong trường hợp không có chồng phổ, bộ lọc đầu ra sẽ lọc các biên tần cao, chỉ để lại tín hiệu tần số cơ bản. Chọn a/b

Đúng

Sai

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Click vào biểu tượng Shape, rectangle

           Description automatically generated để kết thúc chương trình. Quay lại tab Projects và thoát khỏi Code Composer. Tắt nguồn điện và tháo các kết nối (dây dẫn) trên bảng mạch.
    1. Ex 2: Các ứng dụng FFT và Tối ưu.

Mô tả ứng dụng FFT dùng trong bài thí nghiệm.

Ứng dụng được sử dụng trong bài tập này có các đặc điểm sau:

* Tín hiệu analog có biên độ không lớn hơn ±0,9 Vpeak khi đưa đến ĐẦU VÀO ANALOG.
* CODEC lấy mẫu ĐẦU VÀO ANALOG với tốc độ lấy mẫu ƒs = 8000 mẫu/s để thu được các khối gồm 64 giá trị dữ liệu (thực).
* Các giá trị dữ liệu thực được xen kẽ bằng các số 0 để tạo ra một vectơ gồm 64 giá trị phức tạp (128 số ở định dạng Re Im xen kẽ).
* Hàm thư viện thực hiện FFT phức tạp 64 điểm trên vectơ để thu được các giá trị miền tần số. Các kết quả trong miền tần số được đặt trong cùng một vectơ theo thứ tự đảo ngược bit.
* Một hàm thư viện khác sắp xếp lại các giá trị miền tần số theo thứ tự thông thường.
* Phổ cường độ của 20 điểm đầu tiên được tính toán và hiển thị dưới dạng biểu đồ thanh trên màn hình LCD.
* CODEC cũng được sử dụng làm bộ tạo tín hiệu tích hợp để tạo ra tín hiệu hình sin có thể dùng để phân tích. Tần số của tín hiệu này được chọn bằng công tắc INPUT DIP.

Chú ý: Chức năng Tăng tốc phần cứng không được sử dụng trong bài tập này.

* + - * 1. Kết nối bo mạch DSP với máy tính bằng cổng USB số 2 trên bo mạch và cổng USB tiêu chuẩn.

Đảm bảo rằng nguồn điện của bảng mạch đã được BẬT và phần mềm Code Composer được cài đặt như được giải thích trong Phụ lục C.

Khởi chạy Trình soạn thảo mã trên máy tính.

* + - * 1. Chuyển dự án Exercise\_4\_2 thành dự án đang hoạt động bằng cách nhấp chuột vào dự án đó trong cửa sổ Project Explorer.

Mở cả hai tệp main.c và interrupt.c của bài tập này.

Nếu cửa sổ Breakpoints không hiển thị, hãy chọn Breakpoints trong menu View thực đơn. Nếu có bất kỳ điểm dừng nào được hiển thị trong cửa sổ này, hãy nhấp chuột phải vào cửa sổ đó và chọn Remove All.

Khởi chạy chế độ Debug bằng cách nhấp vào biểu tượng A picture containing text, microscope, engine, device

Description automatically generated để tải chương trình vào DSP.

Nếu bạn thấy cảnh báo “shift count is too large”, bạn chỉ cần bỏ qua cảnh báo này.

* + - * 1. Nhấn nút chạy  để chạy chương trình trên DSP. Tại thời điểm này, thông tin sau sẽ hiển thị trên màn hình LCD của bo mạch thí nghiệm là:

LabVolt 91031 Ex4-2

FFT and Optimization

Ứng dụng máy phát tín hiệu hình Sin.

Ứng dụng này bao gồm một bộ tạo tín hiệu có thể tạo ra tín hiệu sóng hình sin ở các tần số khác nhau. Tín hiệu này có sẵn ở ĐẦU RA ANALOG của CODEC. Phần này hướng dẫn cách sử dụng bộ tạo tín hiệu.

* + - * 1. Thực hiện các kết nối như trong Hình. Với những kết nối này, tín hiệu sóng hình sin được tạo sẽ được áp dụng cho đầu vào của Khối khuếch đại âm thanh và máy hiện sóng.
        2. Hình minh họa cách sử dụng công tắc INPUT DIP trên GIAO DIỆN I/O để đặt tần số của bộ tạo sóng hình sin.

Tần số của sóng hình sin được tạo ra bằng giá trị thập phân của cài đặt công tắc DIP nhân với 7,8125 Hz. Mỗi bit được đặt thành 1 sẽ thêm giá trị tần số như thể hiện trong Hình. Tần số thu được là tổng các tần số tương ứng với các bit 1.

Ví dụ: để điều chỉnh tần số thành 250 Hz = 32 x 7,8125 Hz, hãy đặt công tắc DIP thành 32 (0010 0000b).

Để điều chỉnh tần số thành 375 Hz = 48 x 7,8125 Hz, hãy đặt công tắc DIP thành 48 = 32 + 16 (0011 0000b).

Để thay đổi tần số, đặt công tắc DIP thành giá trị mong muốn và nhấn nút ngắt INT0.

Đặt công tắc INPUT DIP thành từng giá trị trong Bảng được cung cấp và xác minh bằng máy hiện sóng rằng bạn thu được tần số mong muốn.

Bạn cũng có thể nghe âm sóng hình sin bằng loa hoặc tai nghe.

Hiển thị FFT.

Ứng dụng này áp dụng FFT cho tín hiệu ở ĐẦU VÀO ÂM THANH để tính toán phổ tần số của tín hiệu. 20 điểm đầu tiên của phổ tần số cường độ được hiển thị dưới dạng biểu đồ thanh trên màn hình LCD.

* + - * 1. Thực hiện các kết nối như trong hình.

Với những kết nối này, sóng hình sin được tạo ra sẽ được đưa đến ĐẦU VÀO ANALOG của CODEC. DSP sẽ tính toán FFT của tín hiệu đầu vào theo thời gian thực và hiển thị phổ tần số cường độ trên màn hình LCD dưới dạng biểu đồ thanh.

Màn hình LCD có thể hiển thị bốn hàng, mỗi hàng 20 ký tự. Trong bài tập này, hai hàng cuối cùng được sử dụng cùng nhau để hiển thị biểu đồ thanh của quang phổ. Vị trí của mỗi thanh trong màn hình tương ứng với tần số trung tâm của một trong các ngăn tần số. Hình minh họa một thanh ở vị trí 2.

Lưu ý: Các vị trí được đánh số từ 0 đến 19.

Hình tiếp theo hiển thị 20 vị trí của biểu đồ thanh và tần số trung tâm của từng ngăn tần số tương ứng. Vị trí 0 tương ứng với thành phần dc của tín hiệu đang được phân tích.

Mỗi thanh trong phổ tần số bao gồm một hoặc hai ký tự khối, như được hiển thị. Chiều cao của mỗi ký tự khối có thể dao động từ 0 đến 8 pixel, nhưng hàng pixel dưới cùng luôn được hiển thị để đánh dấu vị trí. Điều này có nghĩa là chiều cao của mỗi thanh có thể dao động từ 1 đến 16 pixel.

Chiều cao của mỗi thanh biểu thị cường độ tương đối của đầu ra của ngăn tần số tương ứng. Thanh cao 1 pixel biểu thị số 0 (không có tần số tương ứng) và thanh cao 16 pixel biểu thị cường độ hiển thị tối đa cho ngăn tần số đó.

* + - * 1. Điều chỉnh bộ tạo sóng hình sin theo các tần số khác nhau và quan sát quang phổ được hiển thị. Bắt đầu với tần số hiển thị ở đây.

Quan sát Đầu ra trong CCS.

* + - * 1. Đặt tần số của sóng hình sin thành 250 Hz (0010 0000b). Trong Trình soạn thảo mã, tạm dừng chương trình bằng nút Dừng Icon

           Description automatically generated.

Nếu cửa sổ Breakpoints không hiển thị, hãy chọn Breakpoints trong menu View. Cửa sổ này phải trống, như được hiển thị.

Trong tệp main.c, tìm bốn chú thích điểm dừng, nhưng chưa đặt điểm dừng:

//Breakpoints #01

//Breakpoints #02

//Breakpoints #03

//Breakpoints #04

Đặt điểm ngắt trên dòng ngay sau //Breakpoint #01 bằng cách nhấp đúp vào bên trái số dòng. Điểm dừng này sẽ xuất hiện trong cửa sổ <look>Breakpoints</look> như được hiển thị.

Số dòng có thể khác với số được hiển thị trong hình.

Nhấp chuột phải vào dòng điểm ngắt trong Hình và chọn Breakpoint Properties để chỉnh sửa tên. Thay đổi tên của điểm dừng này thành Breakpoint #01 và nhấp vào Edit Property.

Trong tệp main.c, tìm ghi chú //Breakpoint #02, đặt điểm dừng trên dòng sau và đổi tên nó. Thực hiện tương tự với //Breakpoint #03 và //Breakpoint #04.

Khi bạn hoàn tất, cửa sổ Breakpoints sẽ giống như Hình này.

Số dòng có thể khác với số dòng trong Hình. Sử dụng tên để xác định điểm dừng.

Trong cửa sổ Breakpoints, hãy sử dụng hộp kiểm để hủy kích hoạt tất cả các điểm dừng ngoại trừ Breakpoint #01, như minh họa ở đây.

Nhấn nút chạy  để bắt đầu thực thi mã. Quá trình thực thi dừng sau chú thích //Breakpoint #01.

Câu lệnh if nằm phía trên chú thích //Breakpoint #01 kiểm tra giá trị của biến flag\_update\_display. Biến này được đặt thành 1 khi dữ liệu miền thời gian mới được lấy từ CODEC. Vectơ fft\_x\_data chứa dữ liệu thu được.

* + - * 1. Có thể xem dữ liệu miền thời gian trong Trình soạn thảo mã. Trong trình đơn Tools, chọn Graph, sau đó chọn Single Time. Thiết lập các thông số như trong hình này. Vì địa chỉ bắt đầu được đặt thành <code>fft\_x\_data</code> và vì cả Kích thước bộ đệm chuyển đổi và Kích thước dữ liệu hiển thị đều được đặt thành 128, nên tất cả 128 giá trị trong vectơ fft\_x\_data sẽ được hiển thị.

Ở đây chúng ta thấy dòng tại đó quá trình thực thi bị tạm dừng và ví dụ về biểu đồ của tín hiệu thu được trong fft\_x\_data.

Biểu đồ hiển thị tất cả 128 giá trị có trong vectơ fft\_x\_data. Trục hoành hiển thị số mẫu và trục tung hiển thị số mẫu nằm trong khoảng từ 0 đến 127.

Lưu ý: Trong Code Composer, số ngoài cùng bên trái trên trục hoành có thể không bằng 0 mặc dù mẫu đầu tiên trong biểu đồ là mẫu số 0.

Hàm được sử dụng trong ứng dụng này để thực hiện FFT sử dụng thuật toán có giá trị phức tạp. Vectơ đầu vào cho hàm này phải bao gồm các giá trị thực và ảo xen kẽ. (Tham khảo các lập luận trong bảng đã thảo luận trước đó).

Tuy nhiên, dữ liệu mà CODEC thu được là có giá trị thực; không có giá trị tưởng tượng. Tuy nhiên, thuật toán FFT có giá trị phức vẫn có thể được sử dụng bằng cách đơn giản làm cho tất cả các giá trị ảo bằng 0. Đây là lý do tại sao mọi mẫu khác trong fft\_x\_data đều bằng 0.

Do đó, 128 giá trị trong fft\_x\_data phải được hiểu là 64 giá trị phức tạp ở định dạng Re-Im xen kẽ: 64 giá trị thực (dữ liệu được CODEC thu thập) xen kẽ với 64 giá trị ảo, tất cả đều là bằng 0.

Lưu ý: Thuật toán FFT có giá trị phức tạp có thể được sử dụng để tính toán phổ tần số của dữ liệu chỉ chứa các giá trị thực. Điều này được thực hiện bằng cách đặt tất cả các giá trị ảo (mỗi giá trị thứ hai trong vectơ đầu vào) về 0. Hậu quả của việc làm cho tất cả các giá trị ảo bằng 0 là vectơ đầu ra của hàm FFT sẽ đối xứng với tốc độ Nyquist. Điều này có nghĩa là nửa sau của vectơ đầu ra sẽ là hình ảnh phản chiếu của nửa đầu. Tuy nhiên, điều này không gây ra vấn đề gì trong ứng dụng của chúng ta vì nửa đầu của vectơ vẫn sẽ chứa nhiều điểm hơn mức có thể hiển thị trên màn hình LCD.

Bảng tiếp theo sẽ diễn giải nội dung bên trong vectơ <code>fft\_x\_data

* + - * 1. Trong thanh công cụ của cửa sổ biểu đồ, nhấp vào nút Graph Properties. để hiển thị lại hộp thoại Graph Properties.

Chỉ có thể vẽ đồ thị các giá trị thực trong fft\_x\_data</code> bằng cách đặt Tăng chỉ số thành 2 và Kích thước dữ liệu hiển thị thành 64, như được hiển thị. Điều này vẽ đồ thị mọi giá trị khác trong vectơ, bắt đầu bằng giá trị thực đầu tiên. Đặt Đơn vị Thời gian thành ms hiển thị thời gian trên trục hoành.

Hình tiếp theo hiển thị biểu đồ với các cài đặt mới. Như dự đoán, chỉ các giá trị thực được CODEC thu mới được biểu thị. Tất cả các giá trị ảo (số 0 được thêm vào) đều bị bỏ qua.

* + - * 1. Nhấp vào cửa sổ biểu đồ Single Time sẽ hiển thị hai con trỏ màu đỏ cũng như các giá trị mà các con trỏ này biểu thị. Có thể sử dụng những con trỏ này để đo chu kỳ của một chu kỳ của dạng sóng, như được hiển thị.

Nhấp vào cửa sổ biểu đồ và cẩn thận di chuyển con trỏ ngang đến mẫu đầu tiên trên trục tung, đồng thời vẫn giữ nút chuột, di chuyển con trỏ dọc đến giao điểm của con trỏ ngang và chu kỳ tiếp theo của dạng sóng. Điều này hiển thị khoảng thời gian gần đúng tính bằng mili giây của dạng sóng.

Xem xét tần số của sóng hình sin, khoảng thời gian quan sát được có đúng như bạn mong đợi không?. Chọn a/b

Có

Không

|  |  |
| --- | --- |
| Kết quả: | a |

So sánh khoảng thời gian T của sóng hình sin với thời lượng tính bằng giây của khoảng thời gian Twin.

|  |  |
| --- | --- |
| Theo công thức: |  |

Thời lượng cửa sổ Twin bằng: Chọn a/b/c







|  |  |
| --- | --- |
| Kết quả: | a |

Số Chu kỳ trong cửa sổ là Twin / 2 bằng? Chọn a/b/c

8

4

2

|  |  |
| --- | --- |
| Kết quả: | c |

Đây có phải là số nguyên không? Chọn a/b

Có

Không

|  |  |
| --- | --- |
| Kết quả: | a |

Dạng sóng trong biểu đồ được hiển thị dưới dạng một đường liên tục. Đây là đồ thị của tín hiệu liên tục hay tín hiệu rời rạc? Để kiểm tra câu trả lời, hãy hiển thị hộp thoại Graph Properties và thay đổi Data Plot Style từ Line thành Bar. Đặt Time Display Unit thành Sample. Chọn a/b

Liên tục

Rời rạc

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Trong cửa sổ Breakpoints, hãy kiểm tra tất cả bốn Điểm ngắt.

Nhấn nút chạy . Các dòng giữa //Breakpoint #01 và //Breakpoint #02 sẽ được thực thi.

Trước //Breakpoint #02, hàm thư viện cfft tính toán FFT của dữ liệu trong fft\_x\_data. Đây là một hàm phức tạp hoạt động trên các giá trị Phức trong miền thời gian N và xuất ra các giá trị Phức trong miền tần số phức N. Hàm này thực hiện phép tính tại chỗ, nghĩa là kết quả được đặt trong vectơ chứa dữ liệu đầu vào. Nội dung ban đầu của vectơ này sẽ bị hủy trong quá trình này.

Do cách thức hoạt động của thuật toán FFT phức tạp tại chỗ nên các giá trị trong vectơ đầu ra không theo thứ tự bình thường. Thay vào đó, các giá trị này được sắp xếp theo thứ tự đảo ngược bit.

Vì mục đích của chương trình này là hiển thị phổ trên màn hình LCD nên các giá trị phức trong vectơ này phải được sắp xếp lại theo thứ tự bình thường. Điều này được thực hiện bằng hàm cbrev. Hàm này sắp xếp lại dữ liệu và đặt kết quả vào một vectơ khác fft\_out.

* + - * 1. Vectơ fft\_out chứa các giá trị miền tần số ở định dạng Re-Im xen kẽ, như được hiển thị.

Để vẽ biểu đồ các giá trị này trong Code Composer, bạn nên vẽ biểu đồ giá trị thực và giá trị ảo riêng biệt.

Trong trình đơn Tools, chọn Graph, sau đó chọn Single Time. Thiết lập các thông số như trong hình mới. Vì các giá trị thực và ảo được xen kẽ nên việc đặt Tăng chỉ số thành 2 và Địa chỉ bắt đầu thành fft\_out sẽ chỉ vẽ đồ thị các giá trị thực.

Ở đây chúng ta có thể xem ví dụ về biểu đồ giá trị thực của thông tin miền tần số.

Hãy nhớ lại rằng biểu diễn miền thời gian của dạng sóng chỉ chứa các giá trị thực (tất cả các giá trị ảo được đặt thành 0). Mô tả tính đối xứng của các giá trị thực của biểu diễn miền tần số. Chọn a/b/c

Các giá trị thực của biểu diễn miền tần số có tính đối xứng đồng đều về tần số Nyquist.

Các giá trị thực của biểu diễn miền tần số có tính đối xứng kỳ lạ về tần số Nyquist

Không có sự đối xứng

|  |  |
| --- | --- |
| Kết quả: | a |

Phân bổ các đỉnh như thế nào?. Chọn a/b/c

Các đỉnh nằm ở ReX[2] và ReX[52].

Các đỉnh nằm ở ReX[12] và ReX[42].

Các đỉnh nằm ở ReX[2] và ReX[62].

|  |  |
| --- | --- |
| Kết quả: | c |

Tần số này tương ứng với tần số nào dựa trên tần số của đỉnh đầu tiên (ngoài cùng bên trái)?. Chọn a/b/c

-250 Hz

250 Hz

200 Hz

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Hiển thị thuộc tính Graph cho biểu đồ này và đặt Start Address (Địa chỉ Bắt đầu) thành fft\_out+1. Với Index Increment được đặt thành 2, điều này chỉ hiển thị các giá trị ảo ImX[ ].

Mô tả tính đối xứng của các giá trị ảo của biểu diễn miền tần số. Chọn a/b/c

Các giá trị thực của biểu diễn miền tần số có tính đối xứng đồng đều về tần số Nyquist

Các giá trị thực của biểu diễn miền tần số có tính đối xứng kỳ lạ về tần số Nyquist

Không có sự đối xứng

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Nhấn nút chạy . Các dòng giữa //Breakpoint #02 và //Breakpoint #03</code> sẽ được thực thi.

Mục đích của ứng dụng này là hiển thị 20 điểm đầu tiên của phổ biên độ trên màn hình LCD. Điều này được tính bằng đoạn mã sau:

Vòng lặp for khởi tạo các biến với chỉ số i và j về giá trị 0.

Mỗi lượt của vòng lặp lấy hai giá trị liên tiếp từ vectơ fft\_out và bình phương chúng:

Vòng lặp lấy giá trị trong fft\_out[j] (một giá trị thực) và nhân giá trị này với chính nó để thu được bình phương. Kết quả được lưu trữ trong biến a. Sau đó j được tăng lên.

Vòng lặp lại lấy giá trị trong fft\_out[j] (một giá trị ảo) và nhân giá trị này với chính nó để thu được bình phương. Kết quả được lưu trữ trong biến b. Sau đó j được tăng lên.

Các biến a và b được cộng lại với nhau và lưu trữ trong fft\_out[i], ghi đè giá trị thứ i ban đầu của biến này vectơ. Vòng lặp sau đó tăng i và lặp lại. Khi tất cả 20 lần được hoàn thành, 20 giá trị đầu tiên của fft\_out, mỗi giá trị chứa bình phương độ lớn của một điểm phổ.

Để vẽ đồ thị 20 giá trị đầu tiên của fft\_out, hãy thay đổi thuộc tính của đồ thị như được hiển thị.

* + - * 1. Ấn nút run  để chạy chương trình. Các dòng lệnh giữa nhãn //Breakpoint #03 và nhãn //Breakpoint #04 sẽ được thực thi.

Hàm sqrt\_16 xử lý 20 giá trị đầu tiên của mảng fft\_out , lấy căn bậc hai của mỗi giá trị và đặt kết quả vào vị trí tương ứng của vector đầu vào. Sau khi dòng này được thực thi, vector đầu vào chứa phổ tần 20 điểm.

Biểu đồ 20 giá trị đầu tiên trong thư mục gốc, sử dụng Graph Properties được hiển thị ở đây. Biểu đồ sẽ phải giống với mẫu như hình vẽ.

* + - * 1. Vòng lặp for sau //Breakpoint #04 sử dụng các giá trị trong mảng gốc để đặt các ký tự khối thích hợp trong mảng ký tự line3 và line4. Các mảng ký tự này được gửi tới màn hình LCD bằng chức năng updateDisplay. Màn hình LCD hiển thị các ký tự khối ở dòng 3 và 4 của LDC để hiển thị phổ.</p>

Nhấn nút chạy  để thực hiện chương trình. Vòng lặp for sau //Breakpoint #04 sẽ thực thi và sau đó màn hình LCD sẽ được cập nhật. (Việc thực thi sẽ dừng sau nhãn //Breakpoint #01.

So sánh biểu đồ gốc với phổ hiển thị trên màn hình LCD.

Lệch phổ

* + - * 1. Xóa tất cả các điểm ngắt rồi nhấn nút chạy  để thực hiện chương trình. Đặt tần số của sóng hình sin được tạo thành 1000 Hz (1000 0000b). Mô tả Biên phổ trên màn hình LCD.

Phổ có các đặc điểm sau. Chọn a/b/c

hai thành phần ở vị trí 2 (250 Hz) và 8 (1000 Hz).

một thành phần ở vị trí 8 (1000 Hz).

một thành phần ở vị trí 7 (875 Hz)

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Tạm dừng chương trình bằng nút Dừng Icon

           Description automatically generated. Kích hoạt Điểm dừng Breakpoint #02, sau đó nhấn nút chạy . Chương trình sẽ dừng sau khi tính toán FFT và sắp xếp lại dữ liệu.

Code Composer cho phép bạn tách các cửa sổ và đặt chúng ở bất kỳ đâu trên màn hình. Tạo hai biểu đồ mới, một biểu đồ có thuộc tính được hiển thị với Địa chỉ bắt đầu được đặt thành fft\_out và một biểu đồ có cùng thuộc tính ngoại trừ Địa chỉ bắt đầu được đặt thành fft\_out+1.

Nhấp chuột phải vào thanh tiêu đề của một trong các cửa sổ biểu đồ và chọn Detach. Di chuyển cửa sổ này sang bên phải của màn hình. Làm tương tự trong cửa sổ đồ thị khác. Sau đó định vị hai cửa sổ để bạn có thể xem nội dung của chúng cùng một lúc. Ở đây bạn thấy một ví dụ.

Tốc độ lấy mẫu ƒs = 8000 mẫu/s và 64 mẫu phức được FFT xử lý cùng một lúc. Xác định khoảng thời gian Twin. Twin = N / ƒs = ?. Chọn a/b/c

8 mili giây

16 mili giây

10 mili giây

|  |  |
| --- | --- |
| Kết quả: | a |

Có bao nhiêu chu kỳ của dạng sóng tín hiệu hình sin vừa với cửa sổ thời gian?

Twin /T = 16 mili giây/1 mili giây = 16

Twin /T = 8 mili giây/10 mili giây = 0,8

Twin /T = 8 mili giây/1 mili giây = 8

|  |  |
| --- | --- |
| Kết quả: | c |

Đây có phải là số nguyên không?. Chọn a/b

Có

Không

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Xóa tất cả các điểm ngắt rồi nhấn nút chạy . Đặt tần số của sóng hình sin được tạo thành 1062,5 Hz (1000 1000b).

Mô tả phổ hiển thị trên màn hình LCD. Chọn a/b

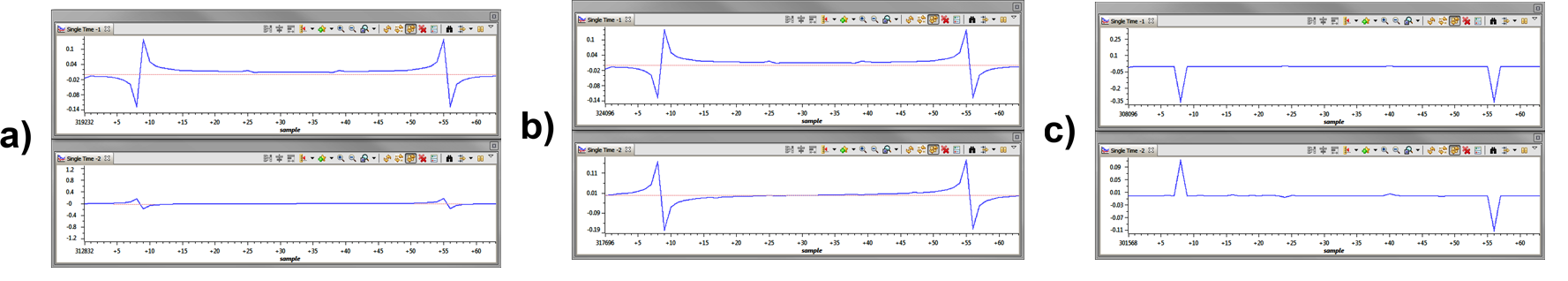
Phổ có thành phần đỉnh ở vị trí 8 (1000 Hz).

Phổ có thành phần đỉnh ở các vị trí 8 (1000 Hz) và 9 (1125 Hz), cũng như các đỉnh ở hai bên.

|  |  |
| --- | --- |
| Kết quả: | b |

Tạm dừng chương trình bằng nút Dừng Icon

Description automatically generated. Kích hoạt Điểm dừng Breakpoint #02 và nhấn nút Chạy . Hai (2) bộ biểu đồ mẫu nào thể hiện chặt chẽ hơn hai biểu đồ miền tần số quan sát được trong Code Composer.



Chọn a/b/c

a. và c.

b. và c.

a. và b.

|  |  |
| --- | --- |
| Kết quả: | c |

Màn hình LCD 1062,5 Hz

Điều gì giải thích các biên trong đồ thị phổ trên màn hình LCD? Chu kỳ của tín hiệu sóng hình sin là T = 1/1062,5 Hz = 0,94 ms.

Số lượng chu kỳ trong cửa sổ Twin/T là = 8,5, không phải là số nguyên.

Khi tần số sóng hình sin tập trung vào một khe tần số (như trong đồ thị c trước đó), chỉ khe đó tạo ra đầu ra. Tuy nhiên, khi tần số nằm giữa hai khe, sẽ có một đầu ra ở một số điểm tần. Điều này là do thực tế là FFT xử lý dữ liệu được lấy mẫu theo thời gian như thể nó là tín hiệu có chu kỳ. Vì số chu kỳ của tín hiệu sóng hình sin trong cửa sổ thời gian không phải là số nguyên nên có sự gián đoạn trong tín hiệu mà thuật toán FFT nhìn thấy. Sự gián đoạn này tạo ra một số lượng lớn các sóng hài xuất hiện dưới dạng “đuôi” trong đồ thị phổ.

Tối ưu.

* + - * 1. Nhìn vào hai dòng theo sau nhãn //Breakpoint #01 trong tệp main.c

Tham khảo phần Thảo luận, hai dòng này gọi những hàm thư viện nào? Chọn a/b/c

cfi và cfft32

cifft và convol2

cfft và cbrev

|  |  |
| --- | --- |
| Kết quả: | c |

Hàm cftt có chức năng gì? Chọn a/b

Nó tính toán iFFT nx điểm phức trên vectơ x, theo thứ tự thông thường. Nội dung ban đầu của vectơ x bị hủy trong quá trình này. Các phần tử phức của nx kết quả được lưu trữ trong vectơ x theo thứ tự đảo ngược bit.

Nó nhận đầu vào là một vectơ chứa nx các giá trị phức trong miền thời gian và thực hiện thuật toán FFT phức tại chỗ, đồng thời các kết quả trong miền tần số được lưu trữ trong cùng một vectơ theo thứ tự đảo ngược bit và ở định dạng Re-Im xen kẽ.

|  |  |
| --- | --- |
| Kết quả: | b |

Hàm cbrev đảo ngược bit các phần tử trong một vectơ phức để đặt các giá trị miền tần số theo thứ tự thông thường. Chọn a/b

Đúng

Sai

|  |  |
| --- | --- |
| Kết quả: | a |

Ưu điểm của việc sử dụng hàm thư viện cho các hoạt động này thay vì lập trình các hàm từ đầu là gì?. Chọn a/b/c/d

Các hàm thư viện là các hàm sẵn sàng sử dụng và có thể rút ngắn đáng kể thời gian phát triển.

Các hàm thư viện là các hàm hợp ngữ được tối ưu hóa cao.

Các hàm thư viện thực thi nhanh hơn các hàm được lập trình bằng ngôn ngữ cấp cao.

Tất cả những điều trên.

|  |  |
| --- | --- |
| Kết quả: | d |

* + - * 1. Tìm từ khoá chỉ dẫn #pragma trong tệp main.c

Chỉ dẫn DATA\_SECTION phân bổ không gian nhớ DATA\_SECTION cho ký hiệu có tên là section. Cú pháp của pragma <code>DATA\_SECTION như sau:

|  |
| --- |
| #pragma DATA\_SECTION (ký hiệu, “tên section”) [;] |

Trong ứng dụng này, tên section là ".input". Trong tệp lệnh trình liên kết <look>lnkx.cmd</look> cho ứng dụng này, một section có tên ".input" được xác định như sau:

|  |
| --- |
| SECTIONS  .input : {} > DARAM0 PAGE 0, align(4) <green>/\* truy cập bộ nhớ mỗi điểm 4 byte \*/ |

Như đã trình bày trong các bảng trước trong Thảo luận , dữ liệu đầu vào cho các hàm thư viện cfft và cbrev phải ở dạng số 32−bit (4 byte). Điều này được thực hiện bằng từ khoá align(4) trong định nghĩa của phần .input. Việc tôn trọng các yêu cầu như thế này có quan trọng không?. Chọn a/b

Đúng

Sai

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Tìm hiểu vòng lặp for sau nhãn //Breakpoint #02 trong tệp main.c.

Trong vòng lặp này, một số hàm được gọi có tên bắt đầu bằng dấu gạch dưới.

Ví dụ: Các hàm \_smpy và \_sadd. Đây là loại hàm gì?. Chọn a/b/c

hàm nội của trình biên dịch

hàm hợp ngữ viết trực tiếp

Đối tượng.

|  |  |
| --- | --- |
| Kết quả: | a |

Lợi ích của việc sử dụng loại hàm này là gì?. Chọn a/b/c

truy cập trực tiếp vào các câu lệnh hợp ngữ.

trình tối ưu hóa có kiến thức tích hợp về cách chúng hoạt động để có thể tối ưu hóa

trình tối ưu hóa có thể mở rộng chúng theo cách khác, căn chỉnh bộ đệm hoặc thực hiện các điều chỉnh khác tùy thuộc vào ngữ cảnh và đối số của lệnh gọi

Tất cả những điều trên

|  |  |
| --- | --- |
| Kết quả: | d |

* + - * 1. Trong cửa sổ C/C++ Projects của Code Composer, nhấp chuột phải vào dự án exercise\_4\_2 và chọn Properties.

Trong cửa sổ Properties của dự án exercise\_4\_2, hãy nhấp vào C/C++ Build. Trong danh sách, bên dưới C5500 Compiler, nhấp vào Basic Options. Trong số những thứ khác, phần này hiển thị mức Tối ưu hóa đã chọn. Mức tối ưu được chọn là gì?. Chọn a/b/c/d

Tối ưu hóa cấp độ 1

Tối ưu hóa cấp độ 2

Tối ưu hóa cấp độ 3

Tối ưu hóa cấp độ 4

|  |  |
| --- | --- |
| Kết quả: | c |

Phân tích tín hiệu ngoài.

* + - * 1. Ta có thể hiển thị phổ của tín hiệu bên ngoài.

Chú ý: CODEC được thiết kế cho tín hiệu điện áp thấp với <b>điện áp đầu vào tối đa là ±0,9 V</b>. Khi sử dụng nguồn tín hiệu bên ngoài, hãy đảm bảo điện áp ở ĐẦU VÀO ANALOG nằm trong phạm vi này. Vượt quá phạm vi điện áp này có thể làm hỏng bo mạch vĩnh viễn.

Tháo cáp kết nối với ĐẦU RA ANALOG</p>

Điều chỉnh độ khuếch đại của Bộ tiền khuếch đại micrô về MIN.

Kết nối tín hiệu âm thanh với ĐẦU VÀO Bộ tiền khuếch đại micrô và kết nối OUTPUT với ĐẦU VÀO ANALOG của CODEC.

Tín hiệu âm thanh có thể đến từ hầu hết mọi thiết bị bao gồm máy tính, máy nghe nhạc mp3 hoặc micrô có mức khuếch đại cao

Trong Code Composer, bỏ chọn tất cả các điểm ngắt. Sau đó nhấn nút chạy 

Đảm bảo có âm thanh ở ĐẦU VÀO ÂM THANH. Tăng từ từ Độ khuếch đại của Bộ tiền khuếch đại micrô cho đến khi bạn nhìn thấy phổ trên màn hình LCD.

Ứng dụng tính toán FFT rất nhanh. Tuy nhiên, mỗi lần cập nhật màn hình LCD sẽ có một chút chậm trễ để đảm bảo hiển thị chính xác. Vì lý do này, đồ thị phổ dường như không thay đổi ngay lập tức.

* + - * 1. Click vào biểu tượng Shape, rectangle

           Description automatically generated để kết thúc chương trình. Quay lại tab Projects và thoát khỏi Code Composer. Tắt nguồn điện và tháo các kết nối (dây dẫn) trên bảng mạch.
  1. U5: Các ứng dụng xử lý tín hiệu
     1. Ex 1: Ứng dụng các bộ lọc FIR và IIR.

Thiết lập và Kết nối mạch

* + - * 1. Thực hiện các kết nối được hiển thị trong Hình.

CODEC được thiết kế cho tín hiệu điện áp thấp có giá trị tối đa là ±0,9 V. Khi sử dụng nguồn tín hiệu bên ngoài, hãy đảm bảo điện áp ở ĐẦU VÀO ANALOG nằm trong phạm vi này.

Vượt quá phạm vi điện áp này có thể làm hỏng bo mạch vĩnh viễn.

* + - * 1. Kết nối bo mạch DSP với máy tính của bạn bằng số cổng USB2 trên bo mạch và cổng USB tiêu chuẩn trên máy tính của bạn.

Đảm bảo rằng nguồn điện của bảng mạch đã được BẬT và phần mềm Code Composer được cài đặt như được giải thích trong Phụ lục C.

Khởi chạy Trình soạn thảo mã (CCS) trên máy tính của bạn.

* + - * 1. Chuyển dự án Exercise\_5\_1 thành dự án kích hoạt bằng cách nhấp vào dự án đó trong cửa sổ Project Explorer

Mở cả hai tệp main.c và interrupt.c của bài tập này.

Khởi chạy chế độ Gỡ lỗi bằng cách nhấp vào biểu tượng A picture containing text, microscope, engine, device

Description automatically generated để tải chương trình vào DSP.

* + - * 1. Trong tệp interrupt.c, hãy tìm phần liệt kê các hệ số bộ lọc. Có bao nhiêu hệ số được liệt kê? Chọn a/b/c/d

16

32

64

72

|  |  |
| --- | --- |
| Kết quả: | c |

Bậc của bộ lọc FIR này là bao nhiêu? Chọn a/b/c

15

63

31

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Nhấn nút chạy  để chạy chương trình trên DSP.

Tại thời điểm này, ta sẽ thấy thông tin trên màn hình LCD của bảng mạch DSP như sau:

LabVolt 91031 Ex5-1

FIR Filter

Thử nghiệm Bộ lọc FIR.

* + - * 1. Sử dụng Máy phát tín hiệu để tạo ra dạng sóng hình sin có biên độ 0,4 V và tần số 100 Hz. Quan sát cả tín hiệu đầu vào (Kênh 1) và tín hiệu đầu ra DSP (Kênh 2) trên máy hiện sóng.
        2. Sử dụng máy hiện sóng để đo biên độ của tín hiệu đầu ra (Kênh 2) theo hàm tần số của tín hiệu đầu vào. Tính hệ số khuếch đại và điền dữ liệu vào bảng.

Hãy nhớ rằng Gain = 20 ∙ log (Aout/Ain) [dB], trong đó Ain = 0,4 V.

Hệ thống sẽ vẽ đồ thị hệ số khuếch đại dưới dạng hàm của tần số.

|  |  |  |
| --- | --- | --- |
| Tần số (Hz) | Điện áp ra (V) | Hệ số khuếch đại Gain(dB) |
| 100 | 0.55 | 2.8 |
| 200 | 0.55 | 2.8 |
| 300 | 0.55 | 2.8 |
| 400 | 0.55 | 2.8 |
| 500 | 0.55 | 2.8 |
| 600 | 0.55 | 2.8 |
| 700 | 0.55 | 2.8 |
| 800 | 0.54 | 2.6 |
| 900 | 0.54 | 2.6 |
| 1000 | 0.54 | 2.6 |
| 1100 | 0.48 | 1.6 |
| 1200 | 0.3 | -2.5 |
| 1300 | 0.1 | -12 |
| 1400 | 0.03 | -22.5 |
| 1500 | 0.03 | -22.5 |
| 1600 | 0.03 | -22.5 |
| 1700 | 0.03 | -22.5 |
| 1800 | 0.03 | -22.5 |
| 1900 | 0.03 | -22.5 |
| 2000 | 0.03 | -22.5 |

Đồ thị: Gain(dB)/Frequency(Hz)

|  |
| --- |
|  |

* + - * 1. Dựa trên biểu đồ thu được, bộ lọc FIR được sử dụng trong ví dụ này là loại bộ lọc nào?. Chọn a/b/c

Bộ lọc thông dải

Bộ lọc thông cao

Bộ lọc thông thấp

|  |  |
| --- | --- |
| Kết quả: | c |

* + - * 1. Tần số cắt của bộ lọc này là bao nhiêu?. Chọn a/b/c

Khoảng 1150 Hz

Khoảng 800 Hz

Khoảng 1350 Hz

|  |  |
| --- | --- |
| Kết quả: | a |

* + - * 1. Kích vào biểu tượng Shape, rectangle

           Description automatically generated để dừng chương trình.

Kiểm tra Bộ lọc IIR

* + - * 1. Chuyển dự án Exercise\_5\_2 thành dự án kích hoạt bằng cách nhấp vào dự án đó trong cửa sổ Project Explorer

Mở cả hai tệp main.c và interrupt.c của dự án này.

Khởi chạy chế độ Gỡ lỗi bằng cách nhấp vào biểu tượng A picture containing text, microscope, engine, device

Description automatically generated để tải chương trình vào DSP.

* + - * 1. Trong tệp interrupt.c, hãy tìm phần liệt kê các hệ số bộ lọc. Có bao nhiêu hệ số được liệt kê? Chọn a/b/c

10 hệ số riêng biệt được liệt kê.

Không có hệ số nào được liệt kê

5 hệ số được liệt kê 2 lần

|  |  |
| --- | --- |
| Kết quả: | c |

Các hệ số được đưa ra theo thứ tự sau: b2 b1 b0 a2 a1

Chúng được viết dưới dạng số tự nhiên nhưng được sử dụng ở định dạng Q31 trong thuật toán. Hệ số a0 được ngầm hiểu có giá trị 1.

|  |  |
| --- | --- |
| Hàm truyền của bộ lọc được sử dụng, đôi khi được gọi là biquad, trông như thế này: |  |

Các hệ số được lặp lại do hàm truyền biquad được sử dụng hai lần (hoặc xếp tầng) để tạo ra bộ lọc IIR cuối cùng. Bậc của P và Q trong phiên bản của hàm truyền là gì?. Chọn a/b/c

P = 2 và Q = 2

P = 2 và Q = 1

P = 1 và Q = 2

|  |  |
| --- | --- |
| Kết quả: | a |

Bậc của bộ lọc IIR hoàn chỉnh (hai bộ tứ liên tiếp) là gì?. Chọn a/b/c

P = 2 và Q = 2

P = 4 và Q = 4

P = 4 và Q = 2

|  |  |
| --- | --- |
| Kết quả: | b |

* + - * 1. Nhấn nút chạy  để chạy chương trình.

Tại thời điểm này, thông tin trên màn hình LCD của bảng mạch DSP như sau:

LabVolt 91031 Ex5-1b

IIR Filter

* + - * 1. Sử dụng bộ tạo tín hiệu để tạo ra dạng sóng hình sin có biên độ 0,4 V và tần số 100 Hz. Quan sát cả tín hiệu đầu vào (Kênh 1) và tín hiệu đầu ra DSP (Kênh 2) trên máy hiện sóng.
        2. Sử dụng máy hiện sóng để đo biên độ của tín hiệu đầu ra (Kênh 2) theo hàm tần số của tín hiệu đầu vào. Tính hệ số khuếch đại và điền dữ liệu vào bảng.

Hãy nhớ rằng Gain = 20 ∙ log (Aout/Ain) [dB], trong đó Ain = 0,4 V.

Hệ thống sẽ vẽ đồ thị hệ số khuếch đại dưới dạng hàm của tần số.

|  |  |  |
| --- | --- | --- |
| Tần số (Hz) | Điện áp ra (V) | Hệ số khuếch đại Gain(dB) |
| 100 | 2.8 | 1 |
| 200 | 2.8 | 1 |
| 300 | 2.8 | 1.2 |
| 400 | 2.8 | 1.6 |
| 500 | 0.51 | 2.1 |
| 600 | 0.53 | 2.4 |
| 700 | 0.55 | 2.8 |
| 800 | 0.55 | 2.8 |
| 900 | 0.52 | 2.3 |
| 1000 | 0.44 | 0.8 |
| 1100 | 0.34 | -1.4 |
| 1200 | 0.24 | -4.4 |
| 1300 | 0.16 | -8 |
| 1400 | 0.09 | -13 |
| 1500 | 0.06 | -16.5 |
| 1600 | 0.04 | -20 |
| 1700 | 0.03 | -22.5 |
| 1800 | 0.03 | -22.5 |
| 1900 | 0.03 | -22.5 |
| 2000 | 0.03 | -22.5 |

Đồ thị: Gain(dB)/Frequency(Hz)

|  |
| --- |
|  |

Dựa trên biểu đồ thu được, bộ lọc IIR được sử dụng trong ví dụ này là loại bộ lọc nào?. Chọn a/b/c

Bộ lọc thông dải

Bộ lọc thông cao

Bộ lọc thông thấp

|  |  |
| --- | --- |
| Kết quả: | b |

Tần số cắt của bộ lọc này là bao nhiêu?. Chọn a/b/c

Khoảng 1450 Hz

Khoảng 800 Hz

Khoảng 1150 Hz

|  |  |
| --- | --- |
| Kết quả: | c |

* + - * 1. Click vào biểu tượng Shape, rectangle

           Description automatically generated để kết thúc chương trình. Quay lại tab projects và thoát khỏi Code Composer. Tắt nguồn điện và tháo các kết nối (dây dẫn) trên bảng mạch.