

Assignment V : FIR Filter Design

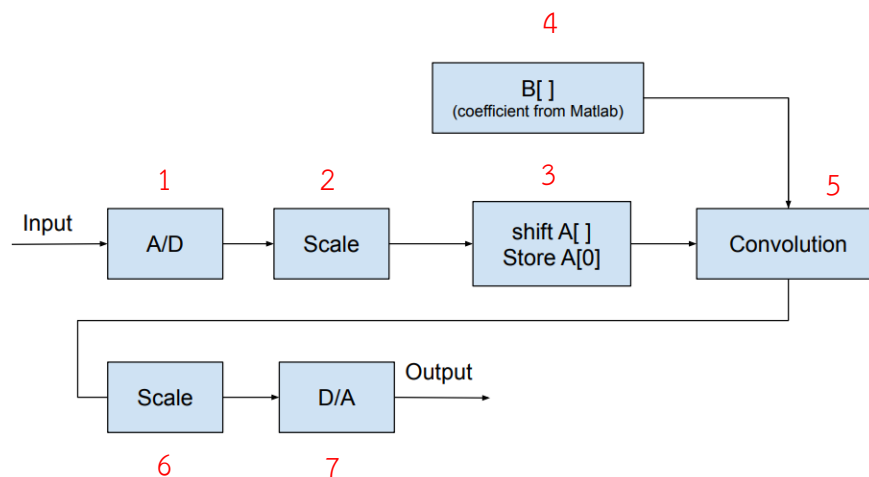
งานมอบหมาย

ให้นักศึกษาต่อวงจร DAC (Digital to Analog Conversion) เข้ากับบอร์ด BBB อาจจะต่อผ่านการ Interface แบบ I2C หรือ SPI ก็ได้ โดย น.ศ. ต้องจัดหาบอร์ดวงจร DAC เอง

เมื่อทำการศึกษาและควบคุม DAC ได้แล้วให้ทำงานดังต่อไปนี้

1. เขียนโปรแกรมควบคุมให้ BBB ผลิตสัญญาณ Sinusoidal แบบมี Offset ออกทางเอาต์พุตของ DAC
2. เขียนโปรแกรมควบคุมให้ BBB ทำงานเป็น FIR Filter (ให้เลือก Sampling rate , filter type, cutoff frequency เอง) โดยให้แสดงวิธีออกแบบอย่างละเอียด ทั้งการคำนวณ/สร้างวงจรกรองด้วย MATLAB และการเขียนโปรแกรม

Procedure



1. A/D รับค่าจากจาก Input ที่เป็นสัญญาณ Analog (Sinusoidal ที่สร้างจาก Wave Generator) เพื่อแปลงเป็น Digital (เข้าสู่บอร์ด BeagleBone Black)
2. ทำการสเกลค่าให้เหมาะสมสำหรับการคำนวณ
3. เพื่อเก็บข้อมูลแบบ Array ทีละตัวให้เราใช้คำสั่งสำหรับ Shift ค่า (เลื่อน) ของ Array A ไปทุกครั้งเมื่อมีค่าใหม่เข้ามา ตามลำดับ เมื่อมีข้อมูลใหม่เข้ามา ให้ทิ้งข้อมูลเดิมทิ้งไป เมื่อ Array มีข้อมูลเกิน
4. ให้ Array B เก็บค่า Coefficient ของ Filter ที่เราได้ Design ผ่านโปรแกรม MATLAB
5. นำค่าจาก Array A Convolution กับ Array B
6. ทำการสเกลค่าให้เหมาะสมเพื่อเตรียมนำค่าออก
7. แปลงค่าจาก BBB ซึ่งเป็นสัญญาณ Digital ให้เป็นสัญญาณ Analog โดยใช้ D/A

Step 1 : Analog to Digital Converter

Assignment 4

CPU ของ BBB เป็นเบอร์ AM335x

ADC: 12-bit SAR ADC
 Sampling rate (max) 200 ks/sec
 Analog channel 8 (กำหนด/เลือกผ่าน analog switch)
 Input Voltage range 0 - 1.8V (ห้ามป้อนเกิน 1.8 V.)
 Input PIN: Port P8 , P9 (ดูในคู่มือ)

จากข้อมูลใน Assignment 4 ทำให้เราทราบว่า Input Voltage อยู่ในช่วง 0 ถึง 1.8 V โดย ADC จะแปลงข้อมูลเป็น 12-Bit ADC นั่นคือ จะมีค่าอยู่ในช่วง 0 ถึง 4095 สำหรับการทดลองจริงในห้องแลป จะป้อนอินพุตเป็น Sinusoidal 1 Vpp offset 0.5 V จะทำให้กราฟมีค่าตั้งแต่ 0 – 1 V

Coding

```
int readAnalog(int num){ //Analog to Digital function
stringstream ss;
ss << LINO_PATH << num << "_raw";
fstream fs;
fs.open(ss.str().c_str(), fstream::in);
fs >> num;
fs.close();
return num;
}
```

Step 2 : Scaling (Normalized Value)

เนื่องจากตัว Input ที่เข้ามายัง BBB เป็น Analog Signal และได้ทำการแปลงเป็น Digital 12-bit แล้วนั้น มี Range ที่กว้างเกินไป คือ 0 – 4095 ดังนั้นเราต้องทำการ Scale หรือ Normalized ให้อยู่ใน Range ที่เหมาะสม ดังแสดงให้ภาพ เพื่อให้มีค่าอยู่ใกล้เคียงกับ Coefficient ของ Highpass FIR Filter

Coding

```
value = (readAnalog(0)/4095.0)-0.5; //Normalized
```

Step 3 : Shift and Store

ทำการ Shift ค่า A[] ทุกครั้งที่มีการเข้ามาที่ A[0] เมื่อไหร่ก็ตามที่ค่าภายใน A[15] และมีค่าใหม่เข้ามา ให้ทิ้งค่าเดิมแล้วทำการเก็บค่าใหม่ไปเรื่อย ๆ ดังแสดง

ADC → A[0] → A[1] → A[2] → A[3] → ... → A[15] → Drop Oldest Value

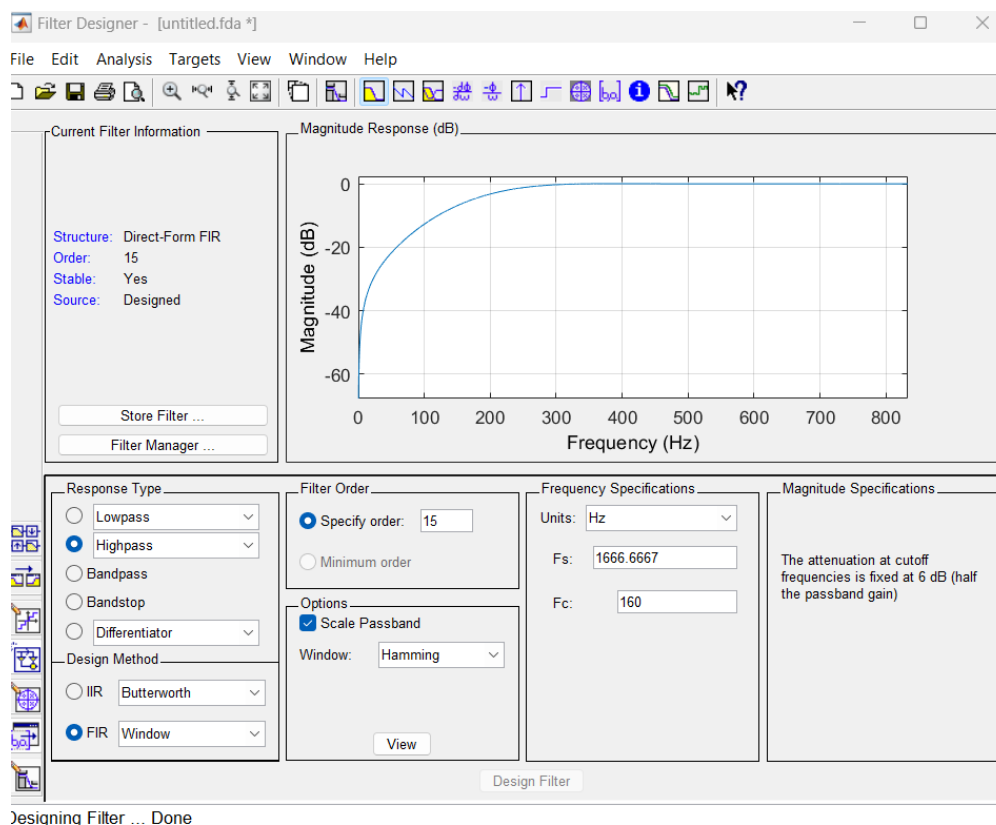
Coding

```
for(int i=15;i>0;i--) // Counting 15 times
{
    A[i]=A[i-1]; // shifting
}
A[0]= ADC; // set Array A[0]= ADC
```

Step 4 : FIR Filter Design by MATLAB

Detail

- Response Type : Highpass filter - Design Method : Window
- Filter Order : 15 - Window : Hamming
- Frequency Specifications : Fs = 1666.6667 Hz และ Fc = 105 Hz



Coefficient : B[16] = { 0.0006363043884066, 0.004173859380219, 0.01323284393, 0.02561597503708, 0.02751786035781, -0.00615679521559, -0.1196116523169, -0.6018576828424, 0.6018576828424, 0.1196116523169, 0.00615679521559, -0.02751786035781, -0.02561597503708, -0.01323284393, -0.004173859380219, -0.0006363043884066 };

Coding

```
float B[16] = {
    0.0006363043884066, 0.004173859380219, 0.01323284393, 0.02561597503708,
    0.02751786035781, -0.00615679521559, -0.1196116523169, -0.6018576828424,
    0.6018576828424, 0.1196116523169, 0.00615679521559, -0.02751786035781,
    -0.02561597503708, -0.01323284393, -0.004173859380219, -0.0006363043884066
}; // Highpass FIR Filter fc = 160 Hz fs = 1667 Hz
```

Step 5 : Convolution

$$Formula : C = \sum_{j=0}^{15} A[j].B[j]$$

ถ้าหากแปลงเป็นโค้ดให้เราทำการนำ Array A[] และ Array B[] มาคูณกันตามลำดับ j แล้วให้นำแต่ละค่าที่คูณกัน มาบวกกัน จะได้เป็นค่า C หรือก็คือผลลัพธ์ของการ Convolution

Coding

```
for(int j=0;j<16;j++) // Counting 16 times
{
    C+=A[j]*B[j]; // Convolution term
}
```

Step 6 : Scaling or Denormalized

ใน Step 2 เราได้ทำการ Normalized ให้กับ A[] เมื่อเราจะนำค่าไปใช้ ต้องทำการ Denormalized ให้เป็นค่าเดิมของมันก่อน ดังนี้

```
}
return (int)((C)+0.5)*200.0); //DeNormalized
}
////////////////////////////////////
```

Step 7 : Digital to Analog Converter

ใช้ DAC ต่อเข้ากับ Output ของบอร์ด BBB โดยที่บอร์ด address เป็น 0x48 ส่งค่าต้องส่งทั้ง Control & Data byte

Coding

```
int main(){ //main function
int x;
cout << "Let start  FIR filter"<< endl;
if((x=open("/dev/i2c-2", O_RDWR)) < 0){
perror("failed to open the bus\n");
return 1;
}
if(ioctl(x, I2C_SLAVE, 0x48) < 0){
perror("Failed to connect to the sensor\n");
return 1;
}
unsigned char writeBuffer[2] = {0};
writeBuffer[0] = 0x40;
while(1)
{
value = (readAnalog(0)/4095.0)-0.5; //Normalized
output = conv(value); //call conv fuction
writeBuffer[1] = output ;
write(x, writeBuffer, 2);
}
close(x);
return 0;
}
```

Overall Coding

```

//Import LIBRARY
#include<iostream>
#include<fcntl.h>
#include<sys/ioctl.h>
#include<linux/i2c-dev.h>
#include<stdint.h>
#include<unistd.h>
#include<math.h>
#include<fstream>
#include<string>
#include<sstream>
using namespace std;
#define LINO_PATH "/sys/bus/iio/devices/iio:device0/in_voltage"
////////////////////////////////////
//Set Parameters and FIR Highpass filter Coefficient
float value,A[16],C;
float B[16] = {
    0.0006363043884066, 0.004173859380219,    0.01323284393, 0.02561597503708,
    0.02751786035781, -0.00615679521559, -0.1196116523169, -0.6018576828424,
    0.6018576828424, 0.1196116523169, 0.00615679521559, -0.02751786035781,
    -0.02561597503708, -0.01323284393,-0.004173859380219,-0.0006363043884066
}; //Highpass FIR Filter  fc = 160 Hz fs= 1667 Hz

int output;
int readAnalog(int num){ //Analog to Digital function
stringstream ss;
ss << LINO_PATH << num << "_raw";
fstream fs;
fs.open(ss.str().c_str(), fstream::in);
fs >> num;
fs.close();
return num;
}
int conv(float ADC) // Convolution function
{
for(int i=15;i>0;i--) // Counting 15 times
{
A[i]=A[i-1]; // shifting
}
A[0]= ADC; // set Array A[0]= ADC
C=0; // set C is zero
for(int j=0;j<16;j++) // Counting 16 times
{
C+=A[j]*B[j]; // Convolution term
}
}

```

```

}
return (int)((C)+0.5)*200.0); //DeNormalized
}
////////////////////////////////////

int main(){ //main function
int x;
cout << "Let start  FIR filter"<< endl;
if((x=open("/dev/i2c-2", O_RDWR)) < 0){
perror("failed to open the bus\n");
return 1;
}
if(ioctl(x, I2C_SLAVE, 0x48) < 0){
perror("Failed to connect to the sensor\n");
return 1;
}
unsigned char writeBuffer[2] = {0};
writeBuffer[0] = 0x40;
while(1)
{
value = (readAnalog(0)/4095.0)-0.5; //Normalized
output = conv(value); //call conv fuction
writeBuffer[1] = output ;
write(x, writeBuffer, 2);
}
close(x);
return 0;
}

```