To use ARM CMSIS-DSP in Keil Studio with MBED 6, follow these steps:

### Step 1: Configure Keil Studio Online IDE

1. Open your web browser and navigate to [Keil Studio Online](https://keil.arm.com/).

2. Sign up or log in to your Keil account.

3. Create a new project and select the STM32L476RG board or any other compatible board from the supported list.

4. Set up your project in the Keil Studio Online IDE.

### Step 2: Include CMSIS-DSP in Your Project

1. In the Keil Studio project explorer, navigate to the "Project" tab.

2. Right-click on your project and select "Manage Library" from the context menu.

3. In the library manager, search for "CMSIS-DSP" and add it to your project.

4. This will include the CMSIS-DSP library in your project, allowing you to use its functions for digital signal processing.

### Step 3: Configure MBED 6 Platform

1. Open the `mbed\_app.json` file in your project.

2. Add the following configuration to specify the use of CMSIS-DSP library:

```json

{

"requires": ["cmsis-dsp"]

}

```

3. Save the `mbed\_app.json` file.

### Step 4: Use CMSIS-DSP Functions in Your Code

1. Include the necessary CMSIS-DSP headers in your code to access the DSP functions. For example:

```cpp

#include "arm\_math.h"

```

2. Use the functions provided by CMSIS-DSP for digital signal processing tasks such as FFT, filtering, etc.

```cpp

// Example: Compute Fast Fourier Transform (FFT) using CMSIS-DSP

arm\_rfft\_fast\_init\_f32(&fft\_instance, FFT\_SIZE);

arm\_rfft\_fast\_f32(&fft\_instance, input\_data, output\_data, 0);

```

### Step 5: Compile and Run Your Project

1. Compile your project by clicking on the "Build" button in Keil Studio.

2. Once the build is successful, download the binary file to your STM32 Nucleo board.

3. Test your application to ensure that CMSIS-DSP functions are working as expected.

### Note:

- Ensure that your STM32 Nucleo board is correctly configured and connected to your computer.

- Refer to the CMSIS-DSP documentation for detailed information on using specific DSP functions.

- Make sure to handle any necessary initialization and configuration for the DSP functions you intend to use in your application.

I apologize for the confusion. In Keil Studio Online, managing libraries is done through a different process. Here's how you can include ARM CMSIS-DSP in your project in Keil Studio Online without using the "Manage Library" option:

### Step 1: Add CMSIS-DSP to Your Project

1. Navigate to the [ARM CMSIS-DSP GitHub repository](https://github.com/ARM-software/CMSIS\_5) and download the latest version of CMSIS-DSP.

2. Extract the downloaded archive to a location on your computer.

### Step 2: Add CMSIS-DSP Files to Your Keil Studio Project

1. In the Keil Studio Online IDE, navigate to the "Project" tab.

2. Click on the "+" icon next to "Files" to add new files to your project.

3. Browse to the location where you extracted the CMSIS-DSP files and add the necessary files to your project. Typically, you'll need the following files:

- `CMSIS/DSP/Include/\*` (Header files)

- `CMSIS/DSP/Source/\*` (Source files)

- `CMSIS/Core/Include/\*` (Core CMSIS header files)

### Step 3: Configure MBED 6 Platform

1. Open the `mbed\_app.json` file in your project.

2. Add the following configuration to specify the use of CMSIS-DSP library:

```json

{

"requires": ["cmsis-dsp"]

}

```

3. Save the `mbed\_app.json` file.

### Step 4: Use CMSIS-DSP Functions in Your Code

1. Include the necessary CMSIS-DSP headers in your code to access the DSP functions. For example:

```cpp

#include "arm\_math.h"

```

2. Use the functions provided by CMSIS-DSP for digital signal processing tasks such as FFT, filtering, etc.

```cpp

// Example: Compute Fast Fourier Transform (FFT) using CMSIS-DSP

arm\_rfft\_fast\_init\_f32(&fft\_instance, FFT\_SIZE);

arm\_rfft\_fast\_f32(&fft\_instance, input\_data, output\_data, 0);

```

### Step 5: Compile and Run Your Project

1. Compile your project by clicking on the "Build" button in Keil Studio.

2. Once the build is successful, download the binary file to your STM32 Nucleo board.

3. Test your application to ensure that CMSIS-DSP functions are working as expected.

### Note:

- Ensure that your STM32 Nucleo board is correctly configured and connected to your computer.

- Refer to the CMSIS-DSP documentation for detailed information on using specific DSP functions.

- Make sure to handle any necessary initialization and configuration for the DSP functions you intend to use in your application.