ReadMe for Pure Assembly Circular Convolution

Procedure to Open the File :

- Download the zip folder and unzip it in whichever location you prefer. (Say in **Documents**)
- Go to **VMware Workstation 14 Player>Windows>Start>Freescale CodeWarrior IDE**
- Once Freescale CodeWarrior IDE opens, go to **File>Open**- Select **Documents>Session3_Dandi_lab1>Pure_Assembly>CircConv5>CircConv5** and click Open.
- In the right-hand side, you ke preceding 'code', and select **CirConv.asm**, and your program should open.

Procedure to Run the File : (Pure Assembly)

- ❏ Go to the **Project** tab, and click **Make**. There shouldn't be any errors.
- ❏ Apply **Breakpoint** at **two** points : Where you declare X_BUF and at rst of the Fmainloop.
- ❏ Go to **Edit>Idm Settings**.. Then under Target Setting Panels, go to **Debugger> Remote Debugging**. Then in Connections Settings, in Connection, in the drop-down menu, select **56800E Simulator.** This is so that we can enable Instruction count/cycle for our program.
- ❏ Again go to the **Project** tab, click **Debug**. You will enter an **Idm.elf thread window**.
- ❏ Go to the **DSP 56800E tab**,which becomes visible only after you debug, and click on **Display Cycle/Instruction count,** and hit **Reset**, so that both become 0.
- ❏ Click the **green triangle** in the Idm.elf thread debug window (**Run**). You will see a blue arrow shift through the code. Keep clicking on the run button till the arrow reaches the last breakpoint (at return 0).
- ❏ Check the Instruction count/cycle by going again to the DSP 56800E tab and clicking on **Display Cycle/Instruction count** as before. You will see the Machine cycles and Machine Instructions simulated in the tab.
- ❏ To check the values out for the variables, go to the **Data** tab, and click on **View Memory**. Type in the variable name whose value's you would want to see, and you will witness the values in hexadecimal.
- ❏ Click on **Debug-Kill** or the **Red cross** in the Idm.elf thread debug window to kill the process. You have to do this every-time you finish debugging.

Functions in Pure_Assembly

Function name	Fasmsum
Input Parameters	Y0 R0
Output parameters	Y0
Assumptions	
Description	This function returns the sum of every value in the given array by adding every value onto the ‘sum’ variable initialized as 0.

Function name	Fasmmultiply
Input Parameters	R3 R2
Output parameters	R4
Assumptions	Both the arrays have to have the same lengths, ‘length’
Description	This function multiplies two arrays in a dot product sense, ie, every value of their respective index positions between the two arrays, and outputs an array.

Function name	Fasmcircflip
Input Parameters	Y0
Output parameters	Y0 Circular Flipped array
Assumptions	None
Description	This function circularly flips the given array of a given length, which means that, barring the first value of the array, the rest of the array is flipped, ie, opposite position values’ across the length are swapped with each other.

Function name	Fasmcircshift
Input Parameters	Y0 R2
Output parameters	Y0
Assumptions	None
Description	This function circularly shifts the array by one value with respect to the given length. It is shifted such that, even after shifting, the length of the array remains the same. This is through the principle of modulas of the index w.r.t the total length. If the index exceeds the length, the value is stored in (i(mod)length) where i is the current index, and ‘length’ is the given length. Thus, evertime this function is invoked, the values shifts by 1 ‘circularly’.