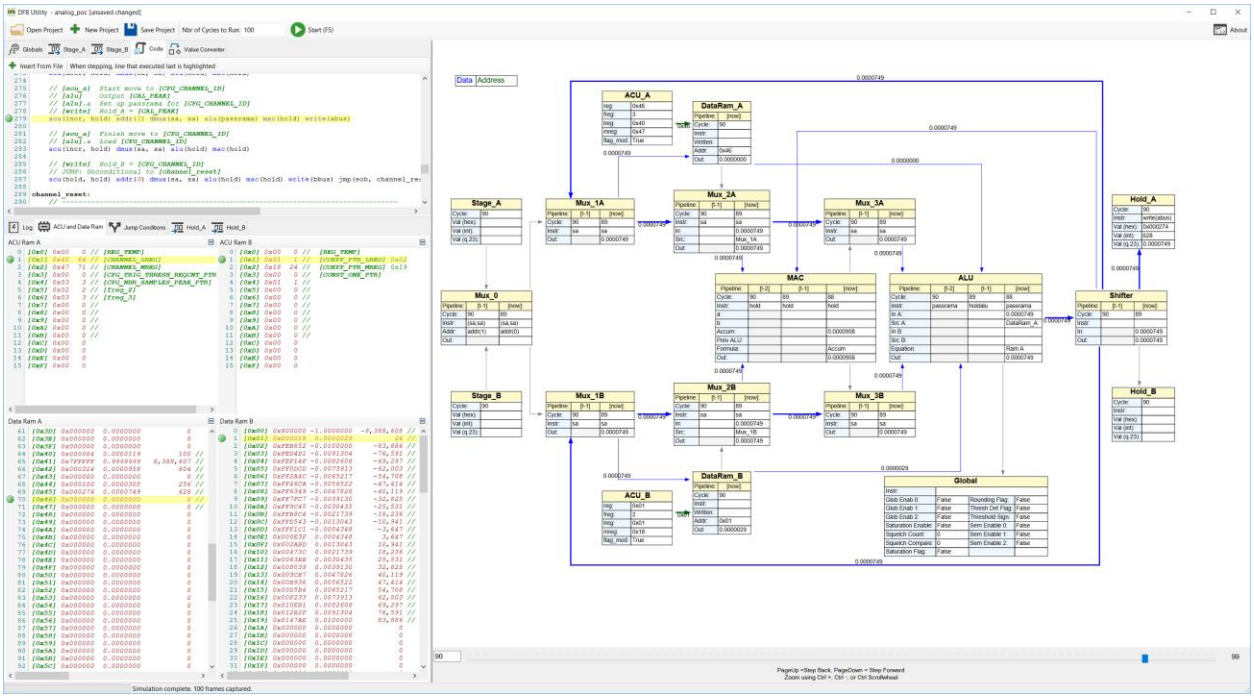
CONTENTS

DFB UTILITY

Version 1.0
Paul Phillips

The DFB Utility is a mini development environment for developing and troubleshooting Cypress PSoC Digital Filter Block assembly code.

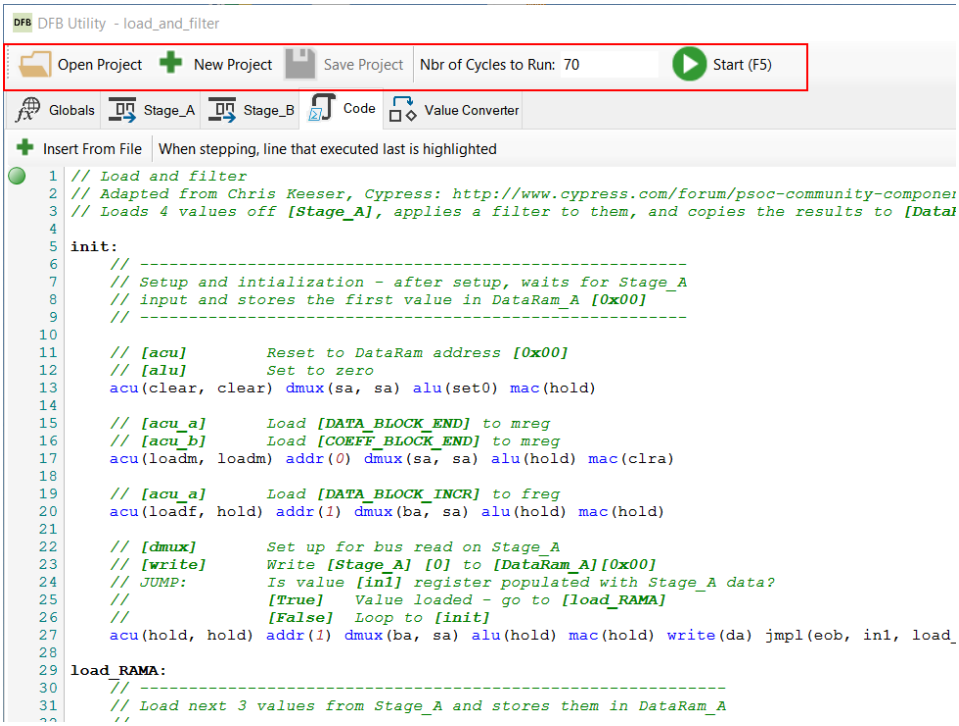
The goal of this tool is to decrease the learning curve and development time of this powerful component by providing a comprehensive view of the machine state and code at each cycle.



GENERAL OPERATION

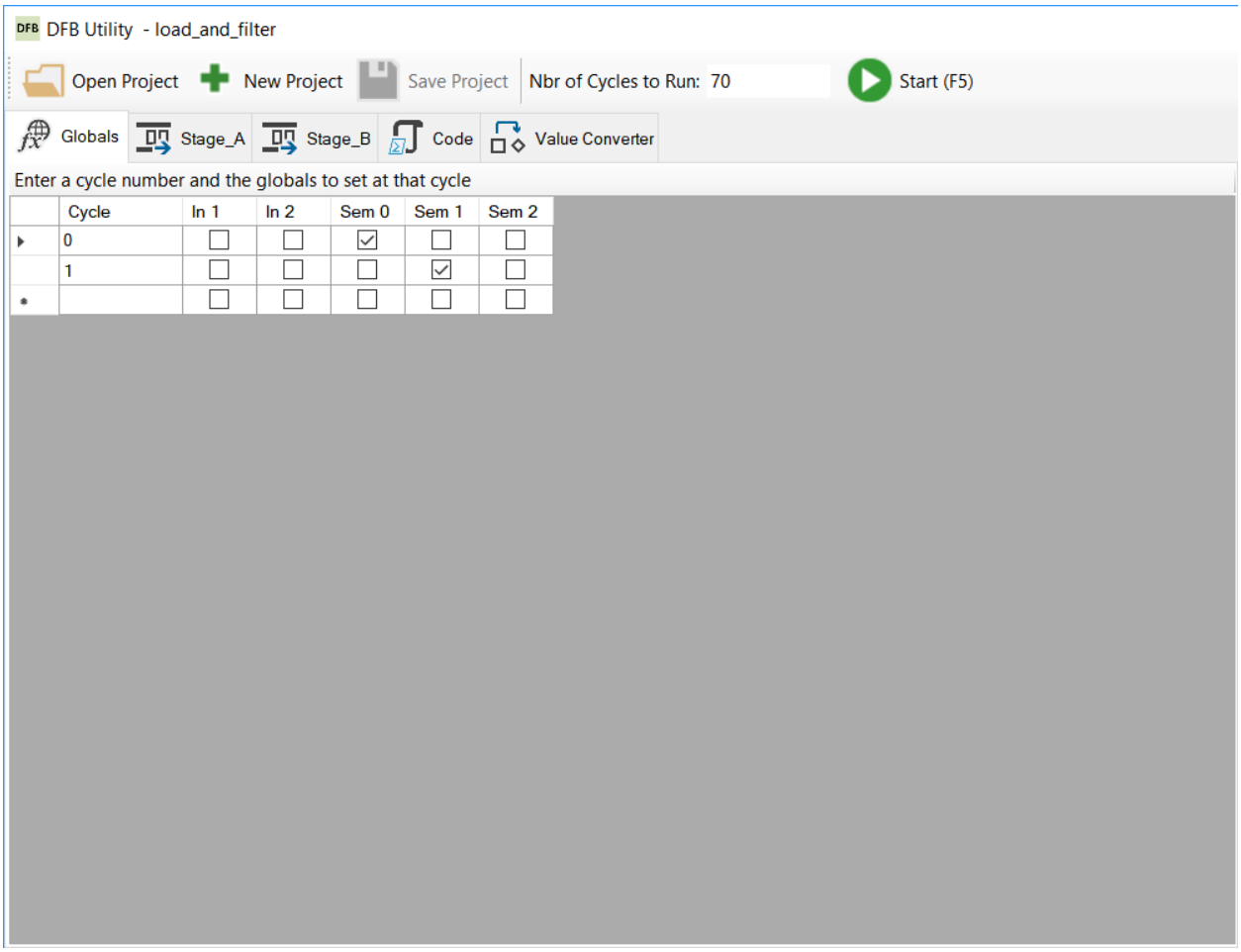
1. Enter Stage Bus In data
2. Populate the Code tab
3. Enter the 'Nbr of Cycles to Run' value
4. Hit F5 or click the Start button
5. Once the simulation and diagram generation are complete, use Page Up and Page Down buttons to 'scrub' through the code either forward or backward
6. Make any code updates and hit F5 to re-assemble. The environment will continue from the same cycle.

MAIN MENU



1. Open Project
    a. Click to open a *.dfbproj file.
    b. *.dfbproj files are simple xml containing the setup information for a project in a single file: Version, Bus1Data, Bus2Data, Code, CyclesToRun, InputSequence
2. New Project - Click to start a new project
3. Save Project – Click to save a projet
4. Nbr of Cycles to Run – enter the number of cycles to run
    a. Note: You may hit enter in this box to start the simulation

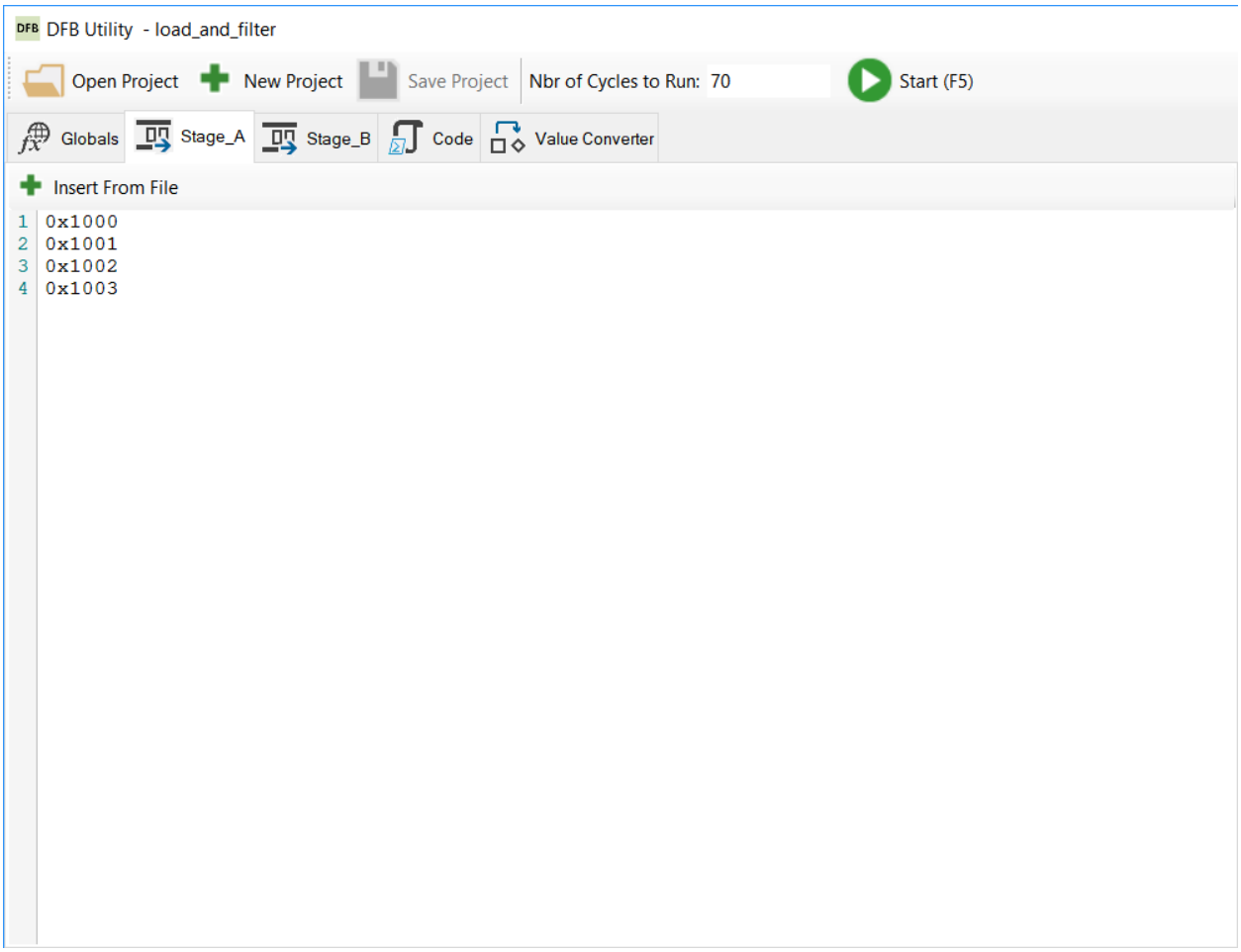
GLOBALS TAB



The globals tab provides a way to set global values for specific cycles. These simulate inputs from the application to the DFB at specific steps

1. Cycle: The cycle number to apply the values
2. In 1 / In 2: Check to set In 1 or 2 to True at the specified cycle
3. Sem 0,1,2: Check to set semaphore 0,1, or 2 to True at the specified cycle

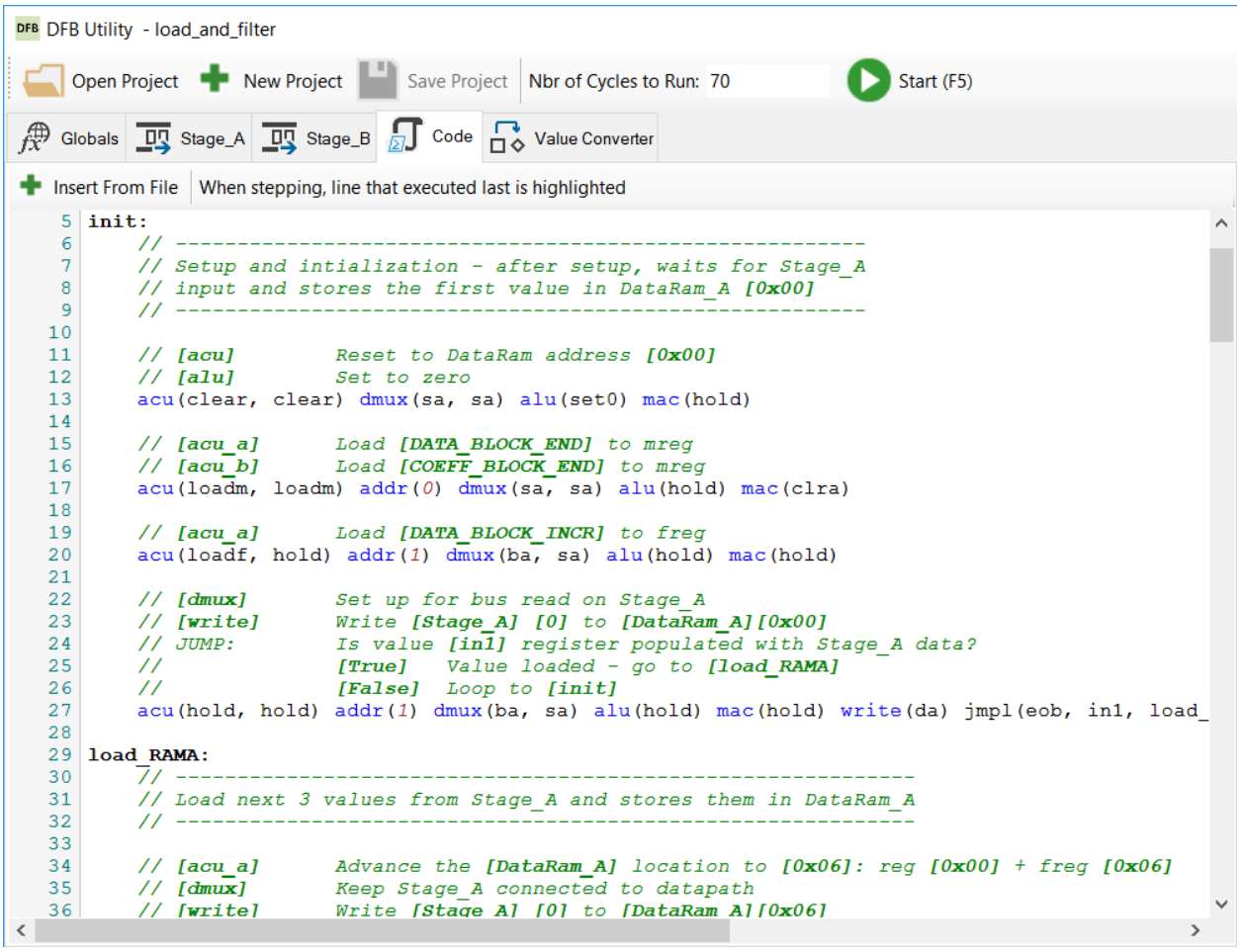
STAGE A / B TABS



These tabs provide simulated input data that will be read by the DFB. Enter values in hex, one line per value.

1. Insert from File: Click this to import the contents of a file into the window. The data will be saved inside the .dfbproj file.

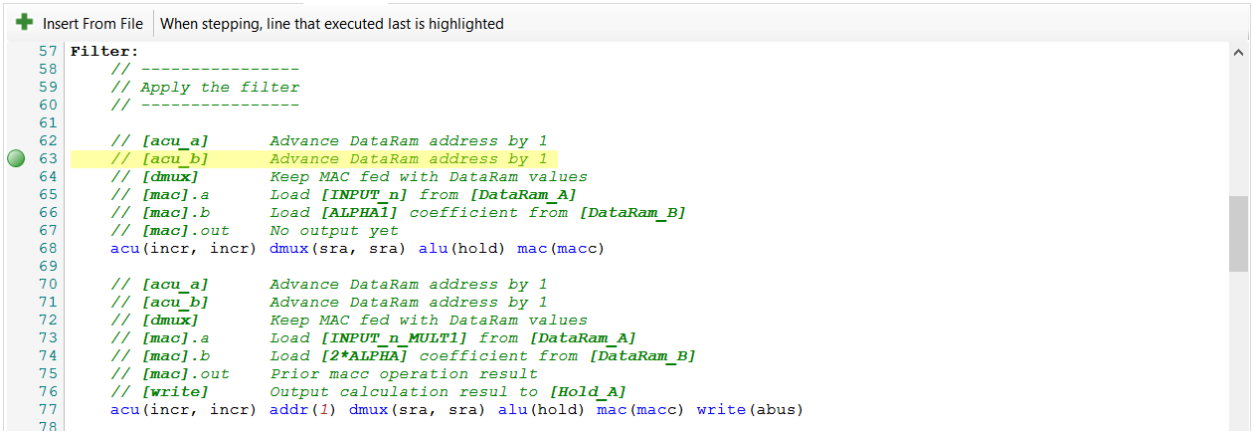
CODE TAB



This tab providex a syntax-highlighted view of the assembler source.

1. Insert from File: Click this to import the contents of a file into the window. The data will be saved inside the .dfbproj file.

As you step forward or backward in the program, the line is highlighted in yellow with a green circle bookmark:

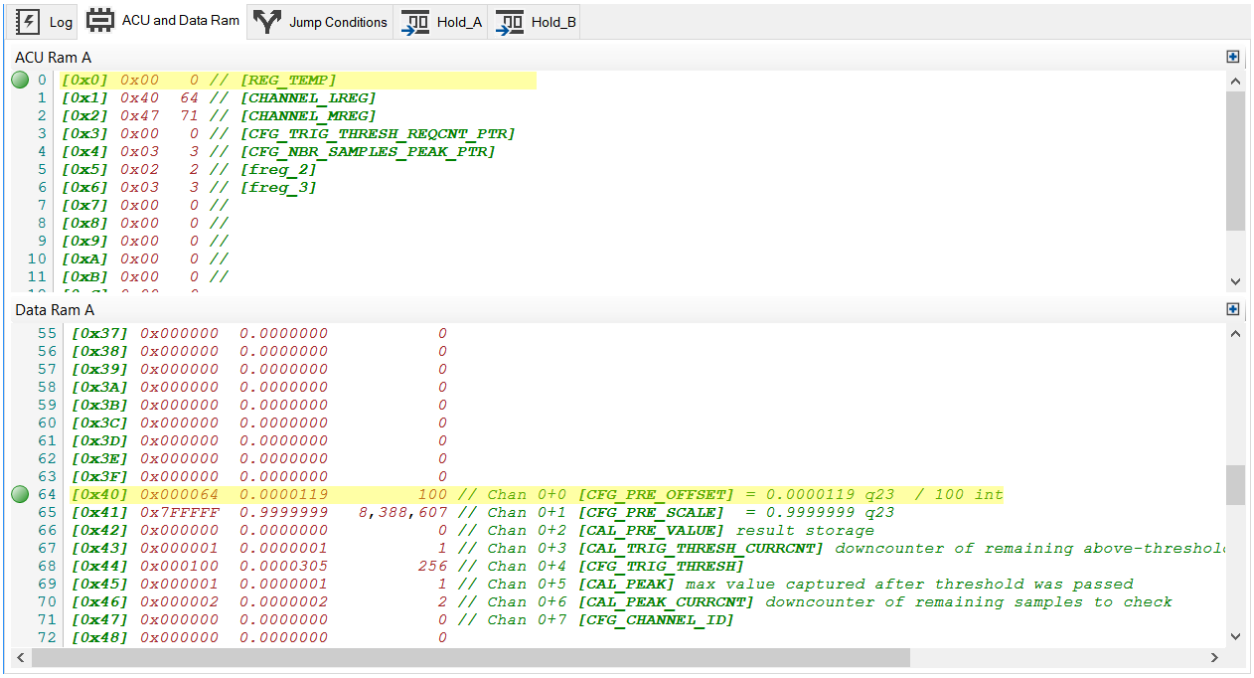


SYNTAX HIGHLIGHTING

1. VLIW keywords are shown in blue
2. Labels are bold
3. Comments are italic green
    a. Text inside square brackets is bolded
4. Data values are shown in magenta

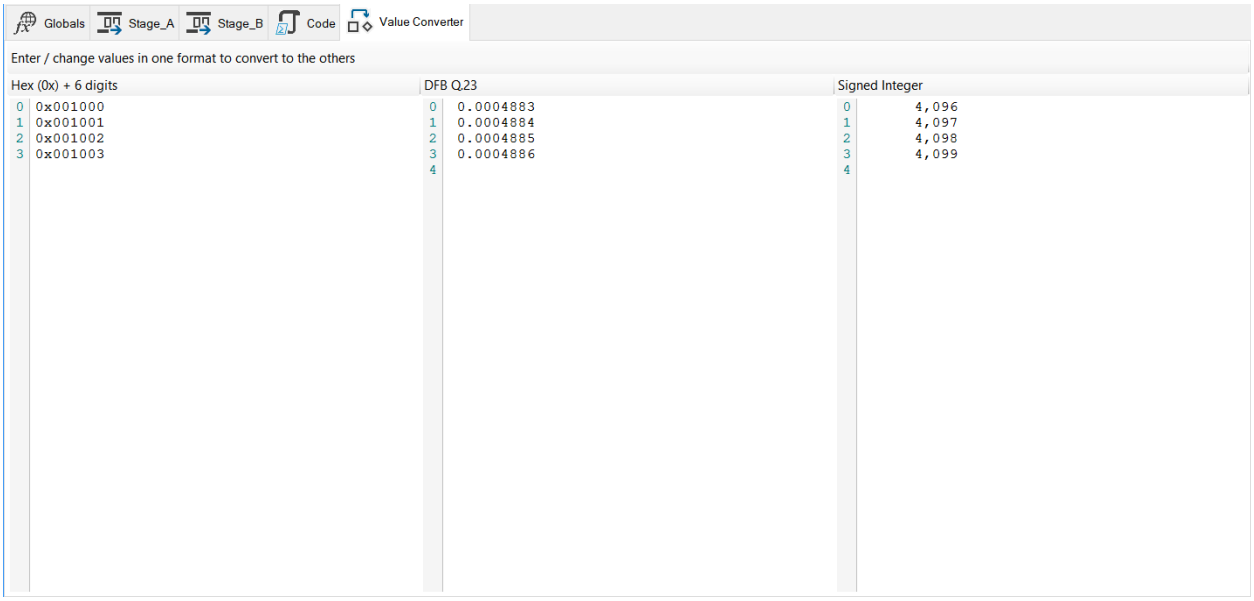
NOTE ON COMMENTS:

It is recommended to add comments next to acu, data_a, data_b entries to tag the purpose of the address, e.g. a variable name. Comments are carried through to the ACU and Data Ram tab so that you can quickly reference the current address's tag as the program executes:



ACU comments are split by using a pipe character, to ensure that the A side and B side appear in the correct window.

VALUE CONVERTER TAB



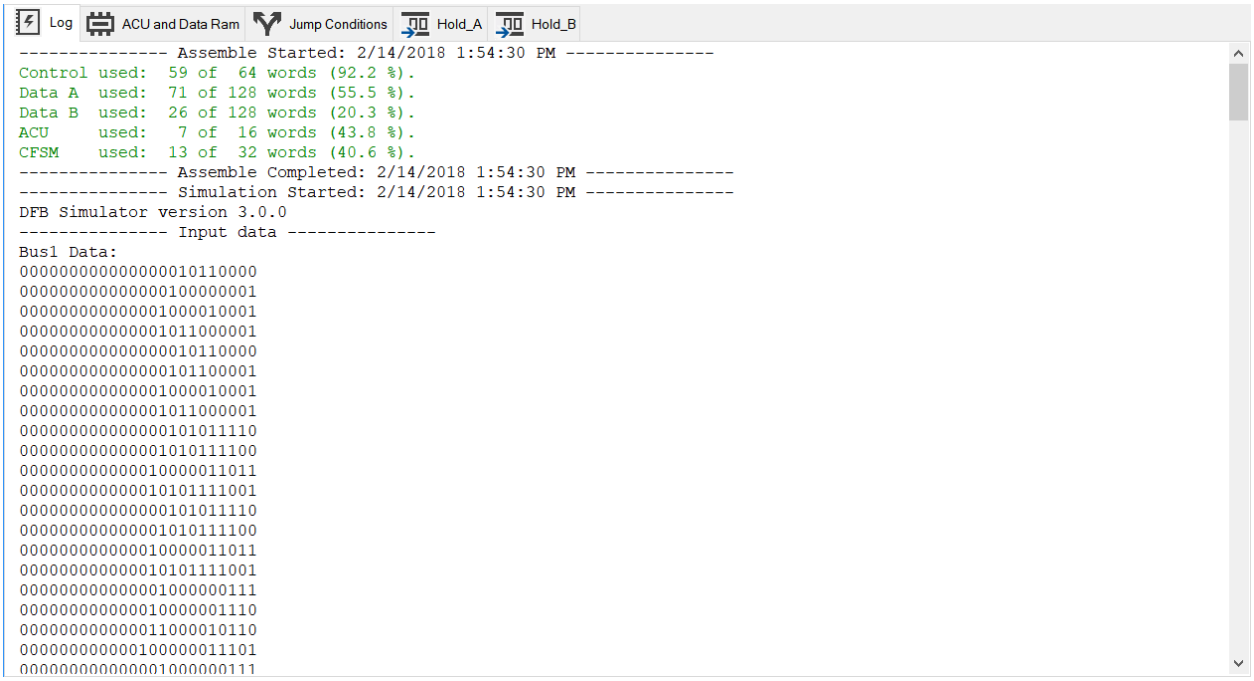
This tab lets you quickly enter a value in one of three formats and have it converted to the others. The DFB uses the following numeric ranges:

DFB Val	Signed Int	Hex Sign Exp.	Hex	Bin					
0.9999999	8,388,607	00007FFFFF	7FFFFF	0111	1111	1111	1111	1111	1111
0.0000001	1	0000000001	000001	0000	0000	0000	0000	0000	0001
0.0000000	0	0000000000	000000	0000	0000	0000	0000	0000	0000
-0.0000001	-1	FFFFFFFFFF	FFFFFF	1111	1111	1111	1111	1111	1111
-1.0000000	-8,388,608	FFFF800000	800000	1000	0000	0000	0000	0000	0000

Type into one of the 3 boxes to enter a value. You may also paste multiple lines into one box.

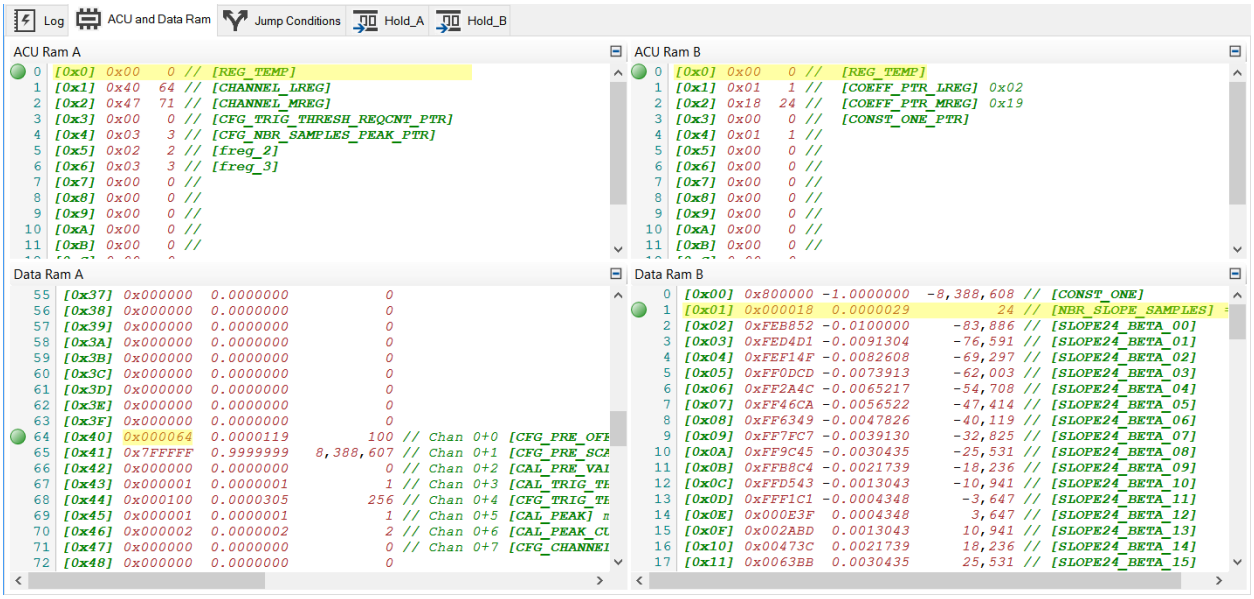
Note: If the converted values do not show, you have either entered a value that is too short or an invalid value for the DFB.

LOG TAB



This tab outputs the log from the assembler component, and also displays any errors that occur.

ACU AND DATA RAM TAB

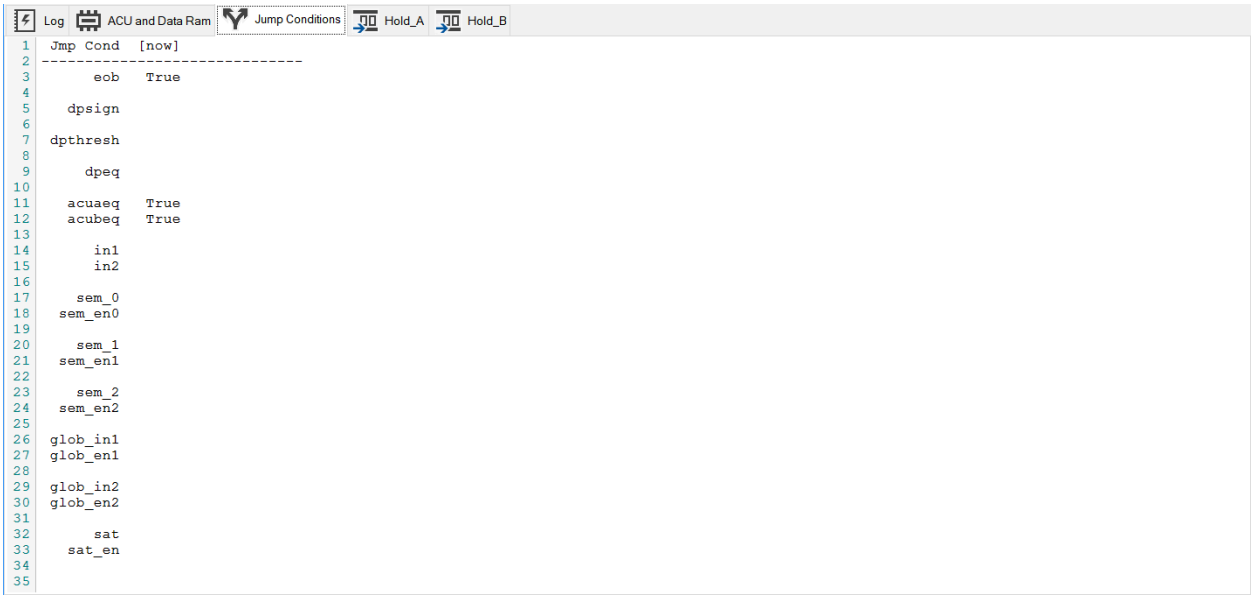


This tab shows the ACU and DataRam values after the current cycle executed. The current address of each is highlighted in yellow with a green circle bookmark.

Either A or B sides may be collapsed.

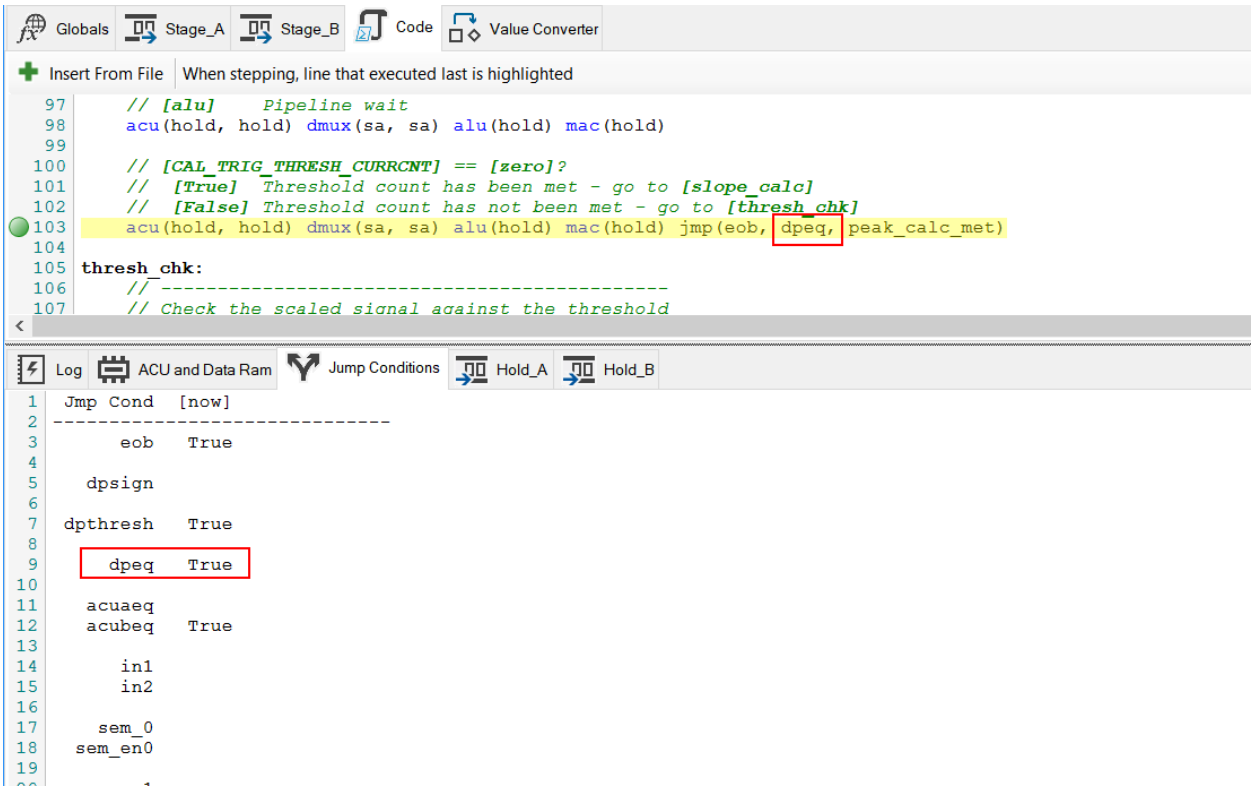
Comments are carried over from the Code as entered.

JUMP CONDITIONS TAB

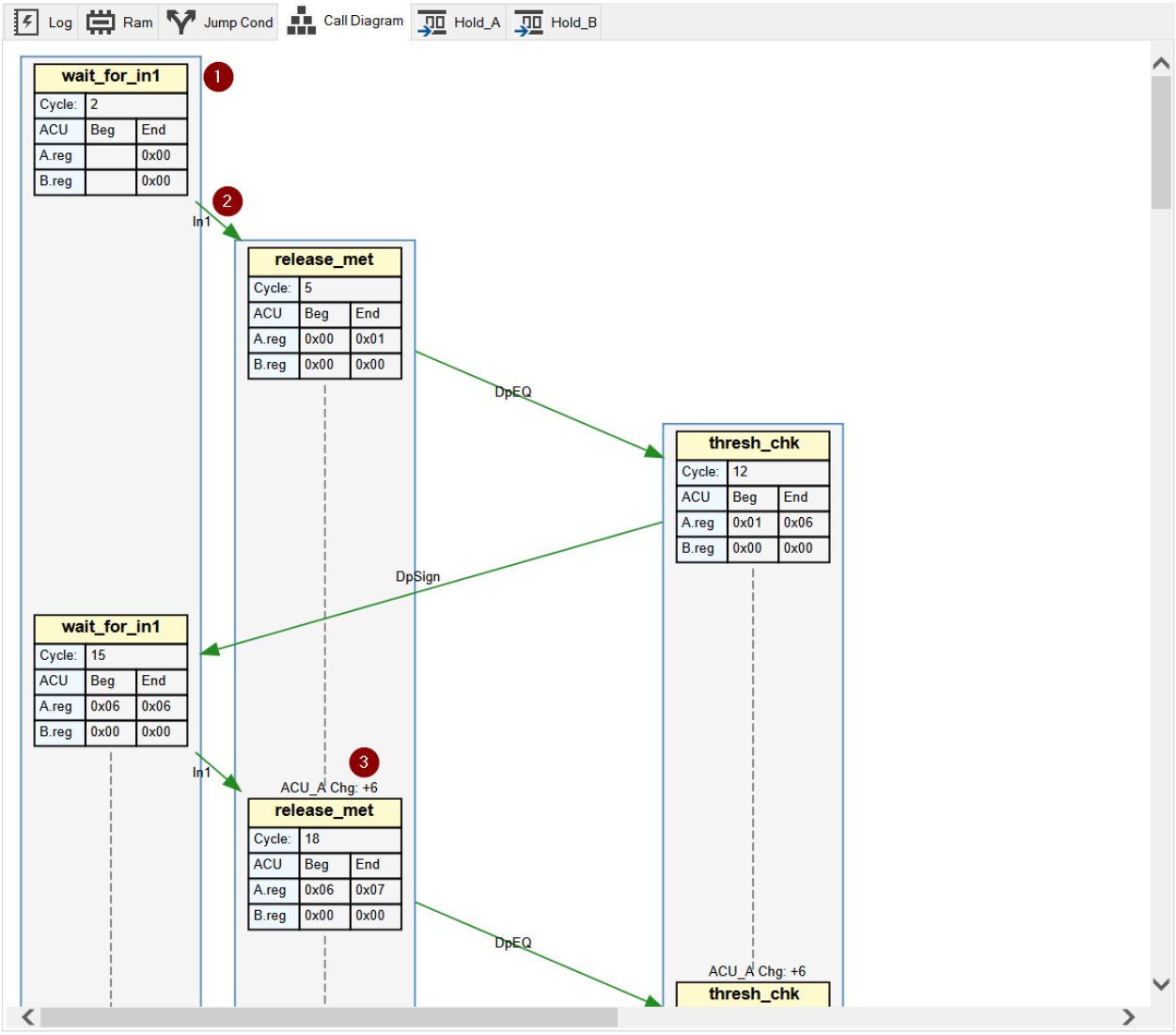


This tab shows the current value of the conditions which may be used as a true branch in jump instructions.

For example, in the below you see that dpeq = True in the jump conditions tab when the code reaches the highlighted line. Dpeq (datapath output equals zero) is the jump condition for true branch, so the code will jump to peak_calc_met instead of falling through to thresh_chk:



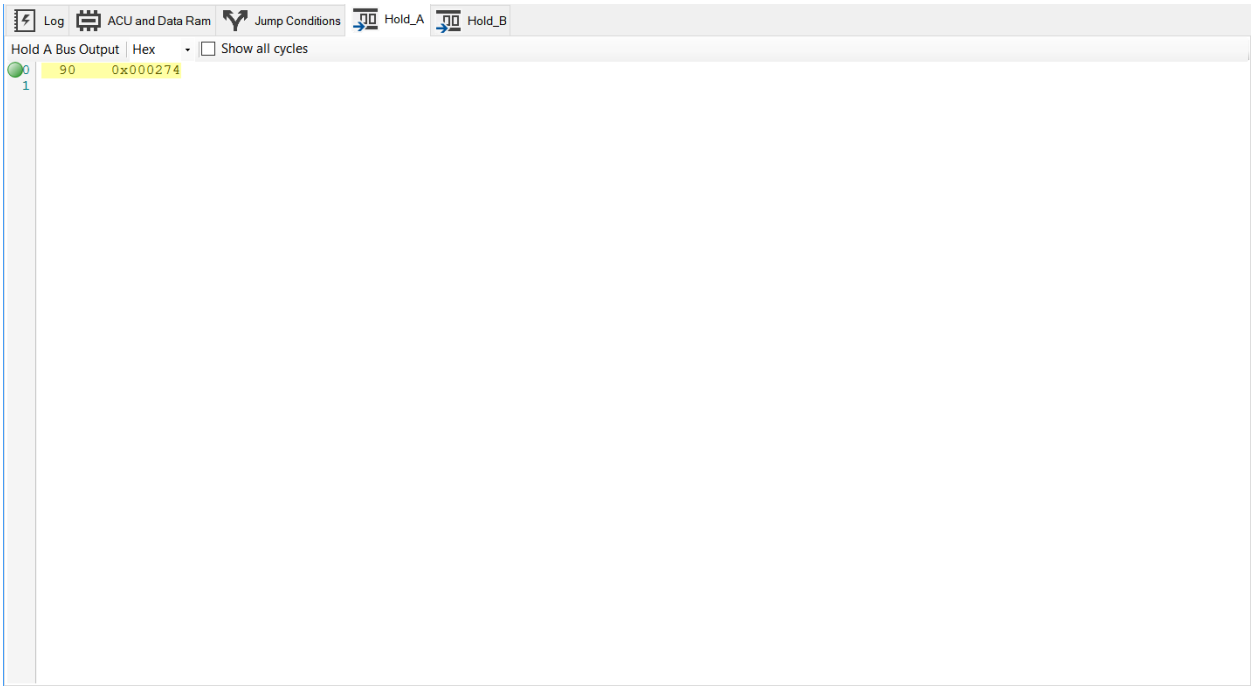
CALL DIAGRAM TAB



The call diagram will show the DFB program's flow.

1. Each label appears in its own vertical lane
    A. ACU address is shown as of the entrance to the label (Beg) and at the label exit (jump instruction - End)
2. Each branch 'true' condition is listed on the line connecting the calls
3. The change in ACU address from one call to the next is displayed
    A. This helps the developer to ensure that the DataRam has been positioned correctly on each label call entry
    B. For example, if the DataRam is organized for multi-channel processing where each channel has 6 coefficients and values, we expect to start a given label at either the start position (e.g. 0x00) , or start position + [6 * current channel number] (e.g. 0x00, 0x06, 0x0C, 0x12, etc)

HOLD A/B TABS

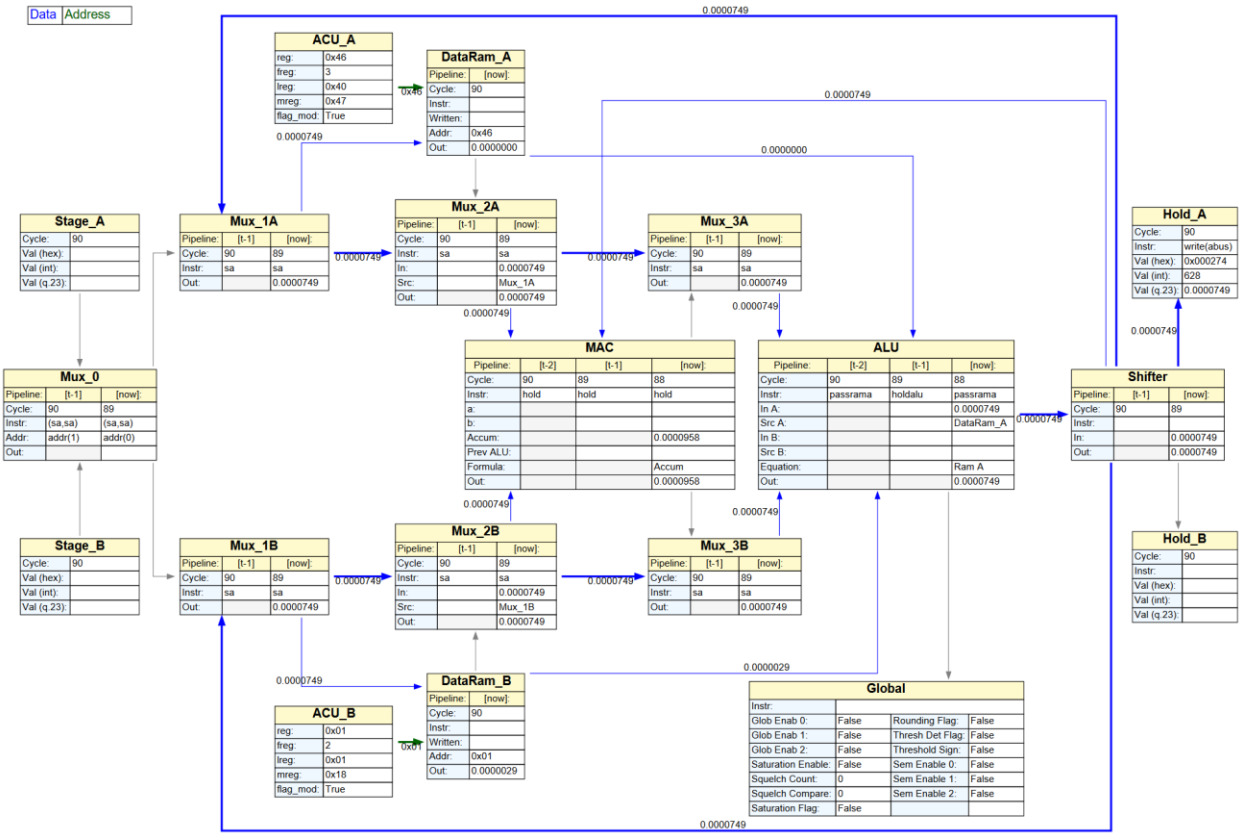


These tabs show the bus out values put into Hold_A and Hold_B. You may choose one of the 3 value formats.

1. Show all Cycles:
    a. Checked: shows all cycles regardless of whether there is an output value
    b. Unchecked: only shows cycles with output values

The current cycle is highlighted in yellow with a green circle bookmark.

DIAGRAM AREA



This tab displays a state diagram for the current cycle.

2. Gray lines indicate physical datapath connections that are not active for the given mux instruction
3. Light blue lines indicate physical datapath connections that are active for the given mux instruction but not being used
4. Bold blue lines indicate physical datapath connections that are active for the given mux instruction and are being used by one of the devices
5. Green lines indicate address buses

Values are displayed in DFB q.23 format for all internal signals. Hex and unsigned integer are shown on Stage and Hold in/out buses.

ZOOM

Use Ctrl + or Ctrl – to zoom in and out of the diagram.

Use Ctrl-0 to reset to default

Changing the window size will zoom the diagram to fit