## Leveraging MATHLIB RTS functions in DSP application

The MATHLIB provides optimized arithmetic, trigonometric and logarithmic functions. These routines are typically used in computationally intensive real-time applications where optimal execution speed is critical. By using these routines instead of the routines found in the existing run-time-support libraries, you can achieve execution speeds considerably faster without rewriting existing code. The MATHLIB library includes all floating-point math routines currently provided in existing run-time-support libraries. These new functions can be called with the current run-time-support library names or the new names included in the MATHLIB library. The library provides both single and double precision floating point functions.

BUILD MATHLIB with OVERRIDE\_RTS Option: (Refer only to rebuild MATHLIB else use prebuilt library)

Default operation of the MATHLIB functions may require users to modify their code to use C, inlined or vector version of the optimized functions. However, there is an option OVERRIDE\_RTS that allows users to take benefit of this library with no modification of the code by replacing links to existing RTS library functions.

This required users to rebuild the MATHLIB with the OVERRIDE\_RTS flag in the build files:

### mathlib\_c66x\_3\_1\_1\_0\packages\ti\mathlib\lib

Locate OVERRIDE\_RTS=0 in the .mk files and replace it with OVERRIDE\_RTS=1 and rebuild the library.

### Prebuilt MATHLIB required to override RTS library functions is provided:

• mathlib\_rts.ae66

#### How do I know if MATHLIB benefits my application?

Check the map file for your application. If the map file indicates that you are using RTS function for any of the trigonometric, logarithmic or arithmetic functions provided by MATHLIB then overriding those functions will give improved performance. Check for following symbols:

\_\_\_\_c6xabi\_divd, sin, cos, log, etc in the .map file for the application. Example:

C:\ti\AM57xx_PRSDK42\ti-cgt	-c6000_8.3	2.2\li	.b\rts6600_elf.lib	)
divd.obj	1536	0	0	
sin.obj	672	0	64	

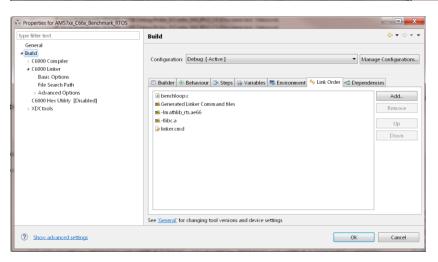
80001720 \_\_c6xabi\_divd 800020a0 sin 800020a0 sinl Steps to leverage MATHLIB in your project:

- 1. Copy mathlib\_rts.ae66 to location\_mathlib\_c66x\_3\_1\_1\_0\packages\ti\mathlib\lib
- 2. In the project, add the library path and the library as shows in screenshot below:

pe filter text	File Search Path	← < ⇒
General		
Build	e e Station (Astron	
▲ C6000 Compiler	Configuration: Debug [ Active ]	<ul> <li>Manage Configurations.</li> </ul>
Processor Options		
Optimization		
Include Options	Include library file or command file as input (library, -l)	🗐 💼 😰 🖓 :
Performance Advisor	"mathlib_rts.ae66"	
Predefined Symbols	"ti.csl.ae66" "libc.a"	
> Advanced Options	nota	
▲ C6000 Linker		
Basic Options		
File Search Path > Advanced Options		
> Advanced Options C6000 Hex Utility [Disabled]	Add <dir> to library search path (search_path, -i)</dir>	🗐 🌒 😵 🖓 :
Could Hex Ounty [Disabled]	"\${SDK_INSTALL_PATH}\mathlib_c66x_3_1_1_0\packages\ti\mathlib\lib" 🚥	
	"\${PDK_INSTALL_PATH}\packages\ti\csl\lib\am 572x\c66\release" ==	
	"\${CG_TOOL_ROOT}/lib" IIII "\${CG_TOOL_ROOT}/include" IIIII	
	s(cG_root_Roots/include	
	Search libraries in priority order (priority, -priority)	
	Reread libraries: resolve backward references (reread libs, -x)	
	✓ Disable automatic RTS selection (disable auto rts)	

3. Go to build options, and link Order Tab as shown below and link math\_rts.ae66 before libc.a which links the RTS libraries:

Properties for AM57xx_C66x_Benchmark_RTOS	hap hate (1) and (30, 90, 90, 12 (2)), and (4) (20, 10)		
type filter text	Build	⇔ • ⇔ • •	
General Processor Options Optimization Include Options Performance Advisor Predefined Symbols > Advanced Options = C6000 Linker Basic Options File Search Path > Advanced Options C0000 Performs = C6000 Linker Utily [Disabled] > XDC tools	Configuration: Debug [Active]	Manage Configurations	Selection Needed       Select objects to add to link order:       Image: Select All         Select All
Show advanced settings	OK	Cancel	OK Cancel



4. You are now ready to rebuild the application. After you rebuild the application recheck the .map file to confirm mathlib is being used in the application by confirming symbols from mathlib are included. Example, map file will show:

80006400	00000080	mathlib_	rts.ae60	5 : divdp.	oe66 (.text:optci)
80007a80 80007aa0	00000020 00000020	mathlib_	rts.ae66		oe66 (.text) oe66 (.text)
C:/ti/math] sindp.oe66 divdp.oe66		1_0/packag 384 160	es/ti/ma 0 0	athlib/lib 0 0	/mathlib_rts.ae66

# **Example RTOS Application with MATHLIB optimization**

Refer to the Example project: AM57xx\_C66x\_Benchmark\_RTOS\_MATHLIB project