

TI BLAS 3.11.0 Delivery

Ming Wei

Where is the BLAS Release

- GIT Server: git.ti.com
- Repository URL: <git@git.ti.com:blas-lib/blas-lib.git>
- TAG: BLAS.03.11.00.01
- This BLAS release is generated based on the modified version of ATLAS 3.11.0. It is available at:

<http://sourceforge.net/projects/math-atlas/files/Developer%20%28unstable%29/3.11.0/>

- The updated ATLAS (most likely 3.12.0) will be available soon

What is in the Package

- blas-lib
 - blas (ECPY version of the BLAS)
 - releasenote.txt
 - example – example code folder
 - inc – header file folder
 - lib – library file folder
 - runscript – C6678 EVM emulation environment setup folder
 - blas_bm (BM version of the BLAS)
 - releasenote.txt
 - example – example code folder
 - inc – header file folder
 - lib – library file folder
 - runscript – C6678 EVM emulation environment setup folder

Testing Environment Setup (1)

- PC with Ubuntu 12.04 LTS
- C6678EVM LE connected to PC via USB
- CCS5.2 installed at /opt/ti/ccsv5
http://software-dl.ti.com/dsps/forms/self_cert_export.html?prod_no=CCS5.2.0.00069_linux.tar.gz&ref_url=http://software-dl.ti.com/dsps/dsps_public_sw/sdo_ccstudio/CCSv5/CCS_5_2_0/
- CGT7.4.2 installed at /opt/ti/TI_CGT_C6000_7.4.2
https://www-a.ti.com/downloads/sds_support/TIcodegenerationTools/index.htm
- XDCTools3.25.02.70 installed at /opt/ti/xdctools_3.25.02.70
http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/rtsc/3.25.02.70//exports/xdctools_setuplinux_3.25.02.70.bin
- “xdctools” is set up as the symbolic link pointing to /opt/ti/xdctools_3.25.02.70
- Bios6.35.4.50 installed at /opt/ti/
http://downloads.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/sysbios/6_35_04_50/exports/bios_setuplinux_6_35_04_50.bin

Testing Environment Setup (2)

- mcsdk_3_00_03_15 installed at /opt/ti
http://software-dl.ti.com/sdoemb/sdoemb_public_sw/bios_mcsdk/latest/exports/bios_mcsdk_03_00_03_15_setuplinux.bin
- omp_2_01_09_00 installed at /opt/ti
- pdk_C6678_2_1_3_7 installed at /opt/ti
- edma3_ild_02_11_10_09 installed at /opt/ti
http://software-dl.ti.com/dsps/dsps_public_sw/sdo_tii/psp/edma3_ild/edma3-ild-bios6/02_11_10_09/exports/edma3_ild_02_11_10_09.tar.gz
- xdais_7_23_00_06 installed at /opt/ti
http://downloads.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/xdais/7_23_00_06/exports/xdais_7_23_00_06.tar.gz
- framework_components_3_30_00_06 installed at /opt/ti
http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/fc/3_30_00_06/exports/framework_components_3_30_00_06.tar.gz
- ipc_1_24_03_32 installed at /opt/ti
http://downloads.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/ipc/3_00_04_29/exports/ipc_setuplinux_3_00_04_29.bin

How to Run Examples (Command Line)

- ECPY example
 - cd blas-lib/blas/example
 - source sourceme.sh
 - make clean
 - make all
 - make run
- BM example
 - cd blas-lib/blas_bm/example
 - source sourceme.sh
 - make clean
 - make all
 - make run

How to Run Examples (CCS 5.2 GUI)

- ECPY example (sgemm)
 - Launch CCS 5.2 GUI
 - Import project from blas-lib\blas\example\Setup_ECPY\app\build\ccs
 - Build the project
 - Load and execute ecpy_example.out
- ECPY example (matrix_vector)
 - Launch CCS 5.2 GUI
 - Import project from blas-lib\blas\example\Setup_ECPY\app\build\ccs
 - Replace sgemm.c with matrix_vector.c
 - Build the project
 - Load and execute ecpy_example.out

How ECPY Example Works

- “make all”
 - First it builds the Setup_ECPY (setup_ECPY.c -> setup_ECPY.obj)
 - It then compiles the sgemm.c and matrix_vector.c with their main() renamed to main_C6678() by using “-Dmain=main_C6678” in Makefile
 - sgemm.obj and matrix_vector.obj will be linked with setup_ECPY.obj, appRman.obj, appEdmaConfig.obj and algIres.obj to make sgemm.out and matrix_vector.out respectively
- “make run”
 - runscript/mmsetup will be called to establish JTAG (XDS-560v2) connection to the C6678 EVM LE
 - runscript/mmruntime will load and execute sgemm.out and matrix_vector.out on C6678 EVM LE one by one
 - The main() in setup_ECPY.c will initialize the cache and the EDMA, then call the main_C6678() to do the matrix operations with ECPY enabled BLAS
 - After the main_C6678() returns, the main() in setup_ECPY.c will release the resources for EDMA and then exit the program
 - runscript/mmterminate will terminate the JTAG connection

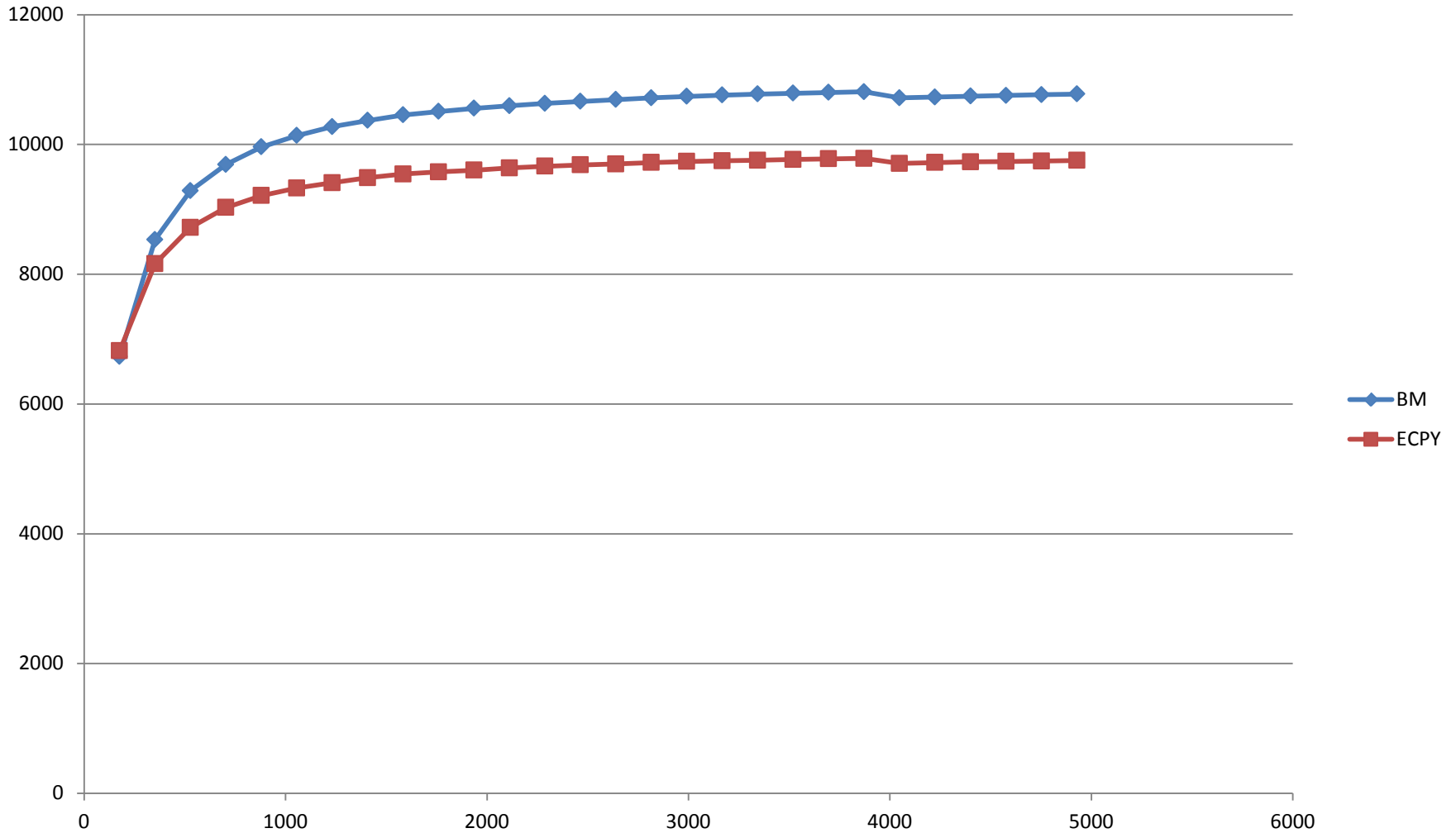
How BM Example Works

- “make all”
 - It builds the sgemm.out, matrix_vector.out and cache_setup.out
- “make run”
 - runscript/mmsetup will be called to establish JTAG (XDS-560v2) connection to the C6678 EVM LE
 - runscript/mmsetup will then load and execute cache_setup.out on C6678 EVM LE to configure the cache
 - runscript/mmruntime will load and execute sgemm.out and matrix_vector.out on C6678 EVM LE one by one
 - runscript/mmterminate will terminate the JTAG connection

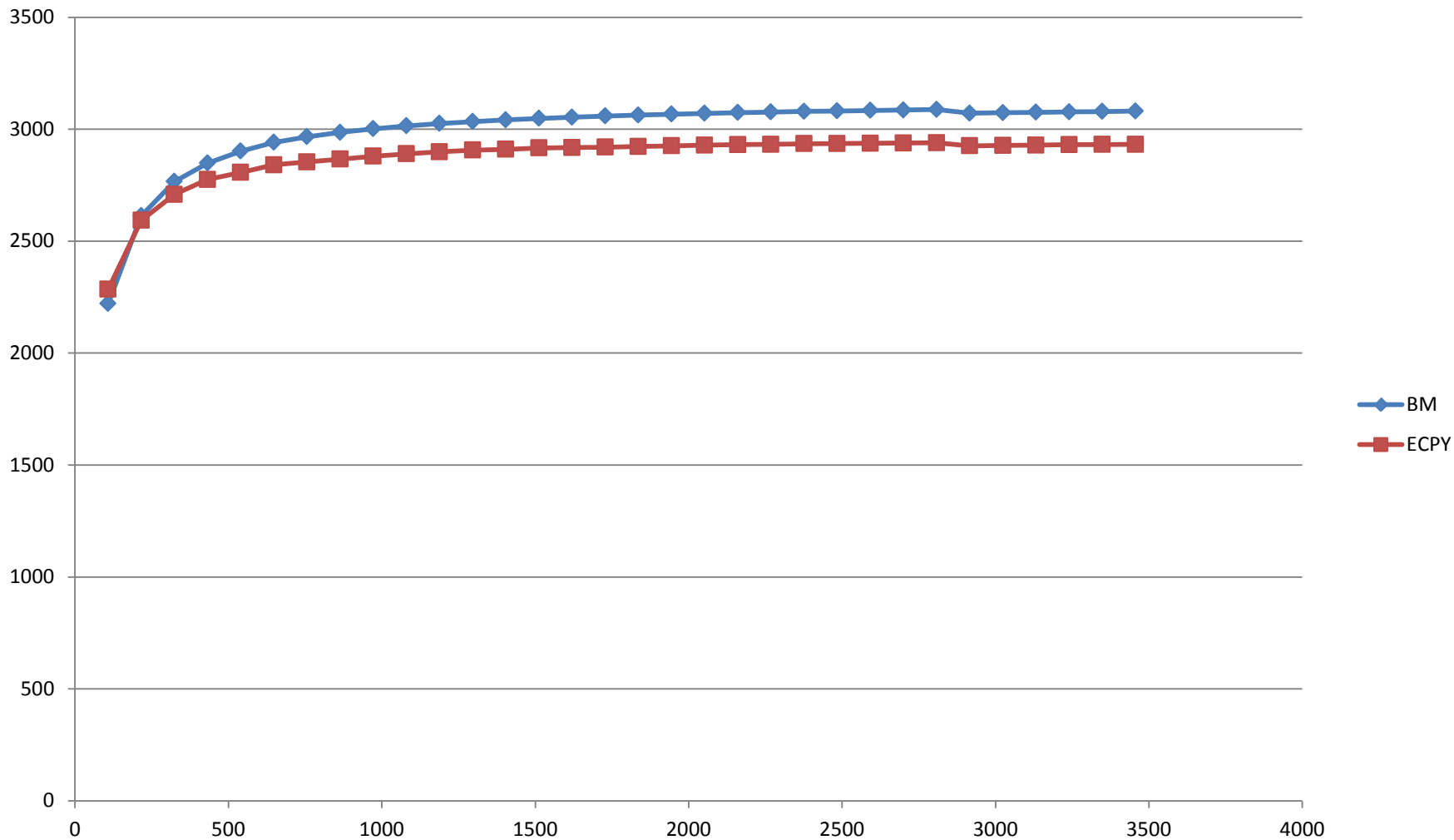
Expected Results

- sgemm expected output:
[367.76, 368.12
674.06, 674.72]
- matrix_vector expected output:
3.0 1.0 3.0
1.0 5.0 9.0
2.0 6.0 5.0
-1.0
3.0
-3.0

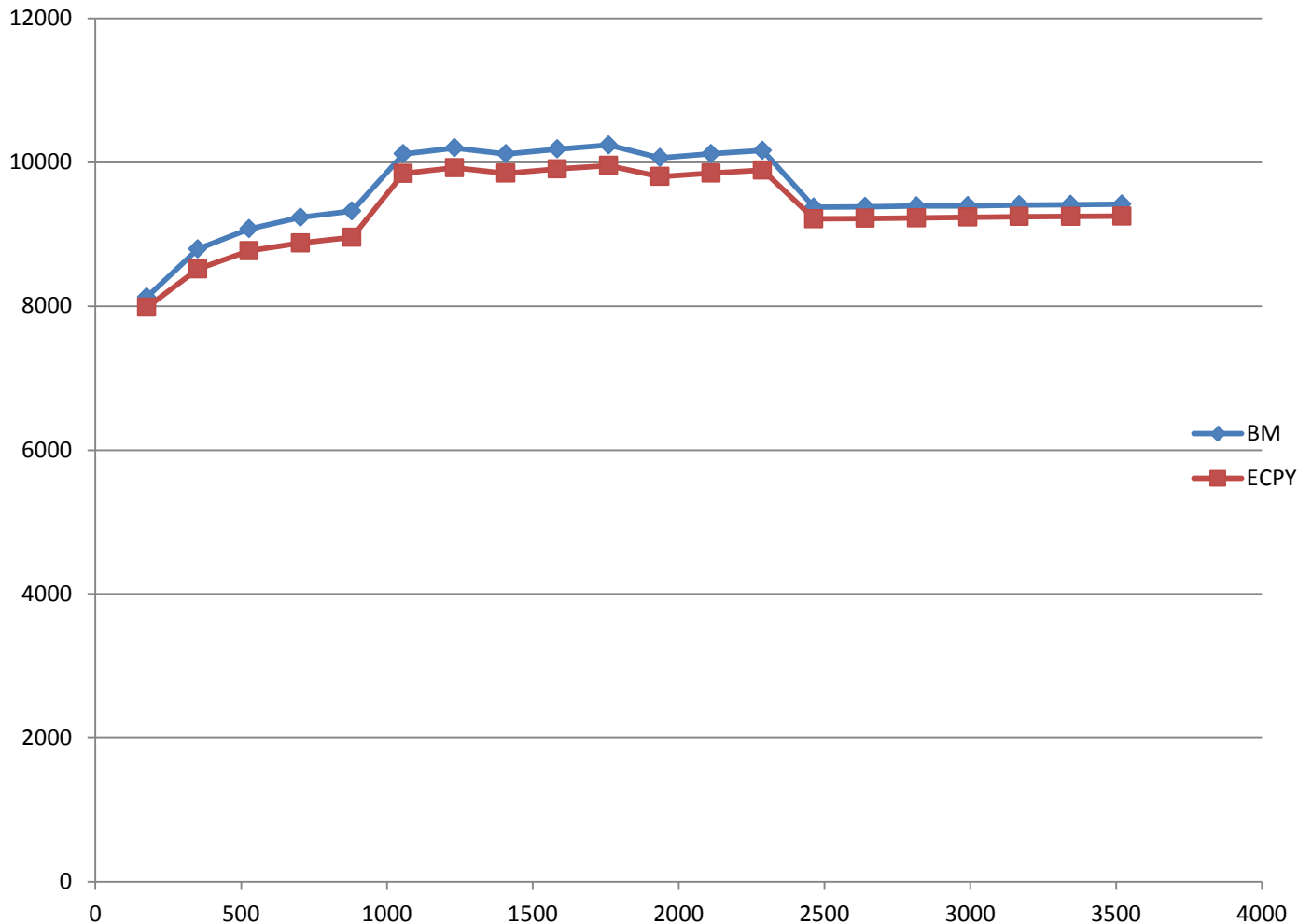
ECPY Performance (Single)



ECPY Performance (Double)



ECPY Performance (Single Complex)



ECPY Performance (Double Complex)

