

Introduction

This document describes the steps for integrating IDT's Servo software with the Cavium Fusion-M evaluation board.

Media

All material referenced in this document is hosted on an IDT Dropbox. For access, please contact support-1588@idt.com. The following table contains a list of files that are hosted in the Dropbox.

Table 1. Media on the FTP Server

File	Description
idt-1588-sw-solution-on-cf-board.pdf	This document
idtConfig.json	JSON configuration file
idtPtp-fusion-m-idt-0-g5ae7c3b	Fusion-M/IDT application binary
idtPtp-fusion-m-pwm-0-g03708aa	Fusion-M/PWM application binary
vmlinux_1645M.64	Linux distribution

Hardware

Table 1 lists the connections between the Fusion-M and the IDT DCO.

Table 2. Connections between Fusion-M and IDT DCO

Board	Connector	Type	Description	Location
Fusion-M	J25	SMA	Not used.	See #20 in Figure 1
	J1	SMA	Clock input to the timestamper.	See #21 in Figure 1
	J4	SMA	PPS input to the timestamper.	
	J2	SMA	PPS output.	
	J5	SMA	Not used.	
	JP8	Header	I2C bus.	
	J11	USB mini	Serial console (USB to serial).	See #7 in Figure 1
IDT DCO	OUT2	SMA	Clock output, connects to J1 on Fusion-M.	See Blue Boxes in Figure 2
	OUT7	SMA	PPS output, connects to J4 on Fusion-M.	
	J3	Header	I2C bus, connects to JP8 on Fusion-M.	N/A

Figure 1. Location of Key Jumpers on Cavium Fusion-M Board



Figure 2. Location of Key Connectors on IDT SMU Board

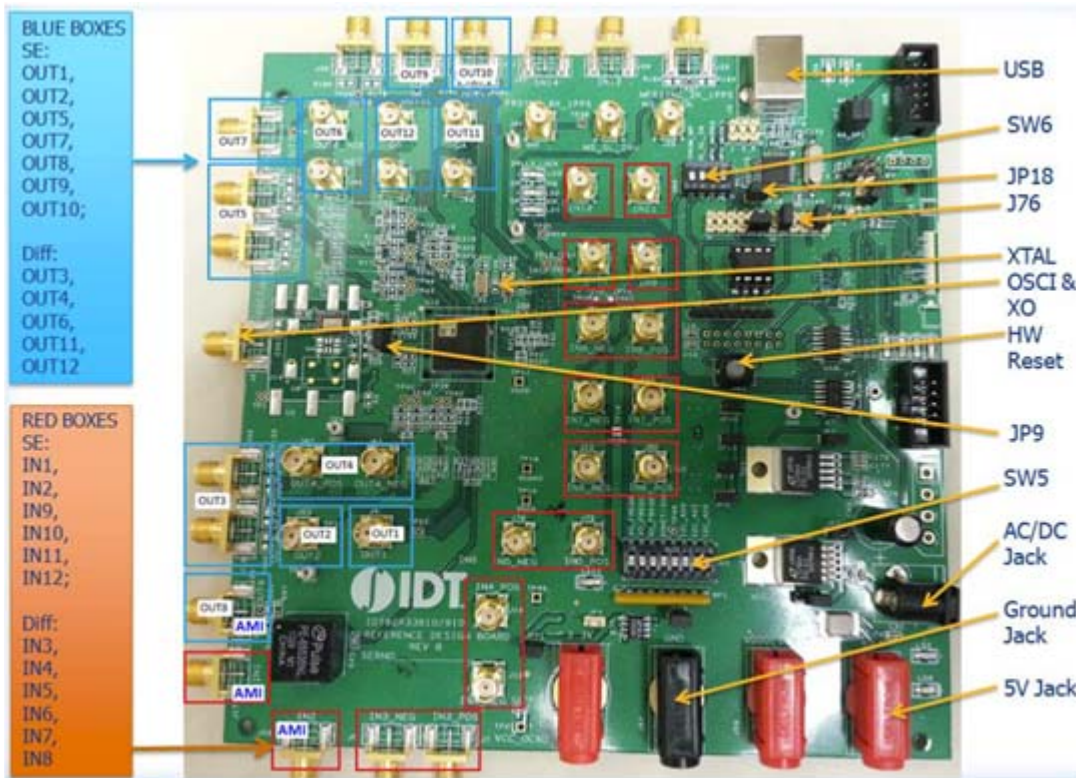
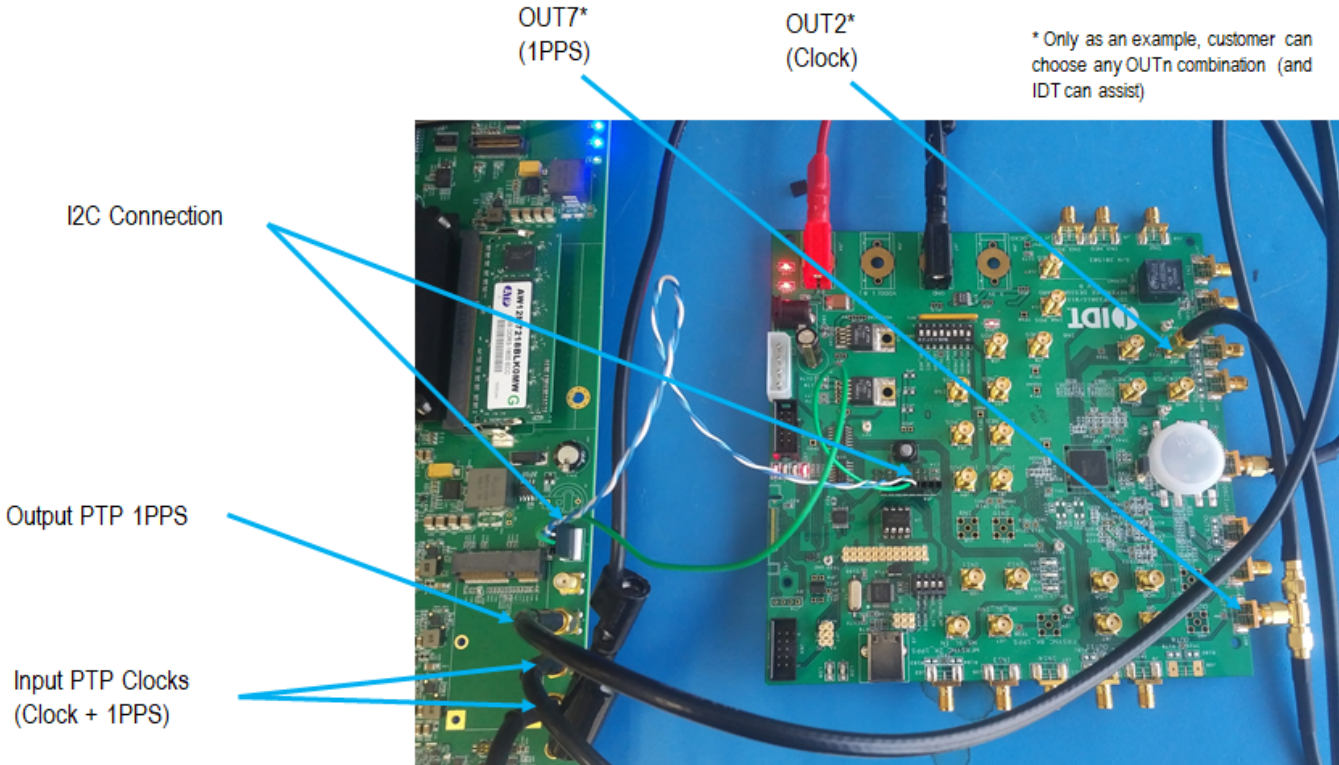


Figure 3. Cavium Fusion-M Board and IDT SMU Board Connected via Cables



Firmware Boot

1. Connect Fusion-M's serial console (J11) to the computer using a USB cable.
2. Open a terminal program such as putty and configure it to 115.2kbps, 1N8, and no flow control.
3. Switch the power on.

You should see the firmware boot sequence on your serial terminal window.

```
Looking for valid bootloader image...
Jumping to start of image at address 0xbfd80000
```

```
U-Boot 2013.07 (Development build, svnversion: u-boot:1594, exec:)-svn1593 (Build
time: Jun 16
```

```
Octeon unique ID: 02800001c919f31e0182 NO.LMC0 Configuration Completed: 8192 MB
Warning: Board descriptor tuple not found in eeprom, using defaults CNF73XX_CRB board revision
major:1, minor:0, serial #:
OCTEON CNF7345-DSP pass 1.3, Core clock: 1200 MHz, IO clock: 600 MHz, DDR clock: 400 MHz (800
Base DRAM address used by u-boot: 0x20f000000, size: 0x1000000
DRAM: 8 GiB
Clearing DRAM..... done Flash: 32 MiB
QLM 4: SGMII
QLM 5: SGMII
MMC:not available
0:PCIe: Port 0 is unknown, skipping. 0:PCIe: Port 1 is unknown, skipping.
PCI console init succeeded, 1 consoles, 1024 bytes each
```

```

Net:cvmx_sfp_read_i2c_eeprom(<NULL>): Error: i2c bus undefined for eeprom
cvmx_sfp_vsc7224_mod_abs_changed: Error reading the SFP module eeprom for <NULL>
cvmx_sfp_read_i2c_eeprom(<NULL>): Error: i2c bus undefined for eeprom
cvmx_sfp_vsc7224_mod_abs_changed: Error reading the SFP module eeprom for <NULL>
octeth0,      octeth1
    
```

4. Type the command "usb start" to scan for USB storage devices.

UBoot

Once at Uboot's prompt, print the environment variables:

```
> printenv
```

```

acsenable=1 atten1=25 atten2=25 autoload=n baudrate=115200
bdk_load=run reset_dlm; run enable_dlm_all; tftp 0x2000000 bdk_cnf73xx_sl... bf=bootoct
$(flash_unused_addr) forceboot numcores=$(numcores) boardname=cnf73xx_crb
boardnum=10005 bootargs=ip=10.64.10.129:::255.255.254.0::eth0:off bootcmd=run i2c_prep; run
linux_tftp_no_dhcp bootdelay=1 bootloader_flash_update=bootloaderupdate
burn_app=erase $(flash_unused_addr) +$(filesize);cp.b $(fileaddr) $(flash_u... bw=20
calpwmreghigh=24016 calpwmreglow=37376 cfgloadby=tftp
e2bootcmd=run mrcm_init;run reset_dlm; run enable_dlm_all; namedalloc ddrp... enable_dlm6=i2c mw
0x21 2 0x71 1; i2c mw 0x21 3 0x71 1;
enable_dlm6_0=i2c mw 0x21 2 0x71 1;
enable_dlm6_1=i2c mw 0x21 3 0x71 1;
enable_dlm6a_7=i2c mw 0x21 3 0xE0 1; i2c mw 0x21 2 0x1D 1;
enable_dlm6b_8=i2c mw 0x21 2 0xE0 1; i2c mw 0x21 3 0x1D 1;
enable_dlm7=i2c mw 0x21 3 0xE0 1; i2c mw 0x21 2 0x8D 1;
enable_dlm7_0=i2c mw 0x21 2 0xC5 1;
enable_dlm7_1=i2c mw 0x21 2 0xA9 1;
enable_dlm8=i2c mw 0x21 2 0xE0 1; i2c mw 0x21 3 0x8D 1;
enable_dlm8_0=i2c mw 0x21 3 0xC5 1;
enable_dlm8_1=i2c mw 0x21 3 0xA9 1;
enable_dlm_all=i2c mw 0x21 3 0x1D 1;i2c mw 0x21 2 0x1D 1; enbctrlrf=0
env_addr=1fbe0000 env_size=20000 eth1ddr=02:68:df:63:39:01 ethact=octeth1
ethaddr=02:68:df:63:39:00 fdtaddr=80000 fileaddr=20000000 filesize=234b450
flash_base_addr=1dc00000 flash_size=2000000 flash_unused_addr=ldf00000 flash_unused_size=ld00000
gain1=35
gain2=35 gateway=10.64.11.254
i2c_prep=i2c dev 1; sleep 1; i2c mw 0x21 2 0xE0 1; i2c dev 0; sleep 1; i2c ... ipaddr=10.64.10.129
linux_mmc=fatload mmc 1 $(loadaddr) vmlinux.64;bootoctlinux $(loadaddr)
linux_tftp_no_dhcp=run mrcm_init;tftp 0x20000000 cavium/experimental/vmlinu...
loadaddr=0x20000000
ls=fatls mmc 0 mk_ubootenv=1
mrcm_init=i2c dev 0; sleep 1; i2c mw 0x25 3 0x00 1; i2c mw 0x25 1 0x07 1; s...
named_block_addr=0x70000000
named_block_size=0x1c713280 netmask=255.255.254.0 numcores=10 numcoreshex=0xa
octeon_failsafe_mode=0 octeon_ram_mode=0 pwdb_in_flash=1 pwmreghigh=24016 pwmreglow=37376
qlm4_mode=sgmii qlm5_mode=sgmii
reset_dlm=i2c dev 1; sleep 1; i2c mw 0x21 2 0xE0 1;i2c mw 0x21 3 0xE0 1; i2...
serverip=10.64.10.70
    
```



```

startapp=1 stderr=serial stdin=serial,pci,bootcmd stdout=serial swloadby=tftp
syslogd_max_size=2048 syslogd_rotated_num=2
uboot_flash_addr=bdd80000 uboot_flash_size=60180000
ver=U-Boot 2013.07 (Development build, svnversion: u-boot:1594, exec:)-svn1...
web_pass=456b7016a916a4b178dd72b947c152b7 web_pass1=a81be4e9b20632860d20a64c054c4150
web_user=admin web_user1=user
    
```

Environment size: 4009/131068 bytes

Network Boot

Connect the board to the network. Make sure the TFTP server, the host, and Fusion-M are on the same network. Serve vmlinux.64 on a TFTP server and make sure the following environment variables are set correctly:

```

> setenv mrcm_init 'i2c dev 0; sleep 1; i2c mw 0x25 3 0x00 1; i2c mw 0x25 1
  0x07 1; sleep 1; i2c mw 0x25 1 0xF8 1; sspi 1:0.19 24 000081 ;sspi 1:0.19 24
  000008 ;sspi 1:0.19 24 010001 ;sspi 1:0.19 24 010201 ;sspi 1:0.19 24 010404
  ;sspi 1:0.19 24 010703 ;sspi 1:0.19 24 010809 ;sspi 1:0.19 24 010900 ;sspi 1
  :0.19 24 010A02 ;sspi 1:0.19 24 010B0C ;sspi 1:0.19 24 0200FF ;sspi 1:0.19
  24 020187 ;sspi 1:0.19 24 020403 ;sspi 1:0.19 24 020809 ;sspi 1:0.19 24 030209
  ;sspi 1:0.19 24 030809 ;sspi 1:0.19 24 030e09 ;sspi 1:0.19 24 031409 ;sspi
  1:0.19 24 031a09 ;sspi 1:0.19 24 032009 ;sspi 1:0.19 24 032100 ;sspi 1:0.19
  24 032309 ;sspi 1:0.19 24 032400 ;sspi 1:0.19 24 032609 ;sspi 1:0.19 24
  032740 ;sspi 1:0.19 24 032D00 ;sspi 1:0.19 24 040218 ;sspi 1:0.19 24 040340
  ;sspi 1:0.19 24 050502 ;sspi 1:0.19 24 050603 ;sspi 1:0.19 24 05070C ;sspi
  1:0.19 24 000F01 ;sspi 1:0.19 24 020301 ;sspi 1:0.19 24 000F01 ;sspi 1:0.19
  24 020300 ;sspi 1:0.19 24 000F01 ;sspi 1:0.19 24 032A01 ;sspi 1:0.19 24 000F01
  ;sspi 1:0.19 24 032A00 ;sspi 1:0.19 24 000F01'
> setenv linux_tftp_no_dhcp 'run mrcm_init;tftp 0x20000000 /path/to/vmlinux.64'
> setenv serverip <tftp server ip address>
> saveenv
    
```

Boot:

```
> run linux_tftp_no_dhcp
```

You should see the kernel log messages followed by the terminal prompt (login is not required).

SD Card Boot

1. Prepare an SD card with a single partition running the FAT file system.
2. Copy vmlinux.64, the idtPtp binary, and the JSON configuration file to that partition.

```
> fdisk -l
> fdisk /dev/mmcblk0
> mount /dev/mmcblk0p1 /mnt/
> cp /path/to/vmlinux.64 /mnt
> (Repeat for idtPtp and JSON file)
```
3. Insert the SD card into the SD card connector on Fusion-M. Make sure the following environment variables are set correctly:

```
> setenv mrcm_init 'i2c dev 0; sleep 1; i2c mw 0x25 3 0x00 1; i2c mw 0x25 1
0x07 1; sleep 1; i2c mw 0x25 1 0xF8 1; sspi 1:0.19 24 000081 ;sspi 1:0.19 24
000008 ;sspi 1:0.19 24 010001 ;sspi 1:0.19 24 010201 ;sspi 1:0.19 24 010404
;sspi 1:0.19 24 010703 ;sspi 1:0.19 24 010809 ;sspi 1:0.19 24 010900
;sspi 1:0.19 24 010A02 ;sspi 1:0.19 24 010B0C ;sspi 1:0.19 24 0200FF ;sspi 1:0.19
24 020187 ;sspi 1:0.19 24 020403 ;sspi 1:0.19 24 020809 ;sspi 1:0.19 24 030209
;sspi 1:0.19 24 030809 ;sspi 1:0.19 24 030e09 ;sspi 1:0.19 24 031409 ;sspi
1:0.19 24 031a09 ;sspi 1:0.19 24 032009 ;sspi 1:0.19 24 032100 ;sspi 1:0.19
24 032309 ;sspi 1:0.19 24 032400 ;sspi 1:0.19 24 032609 ;sspi 1:0.19 24
032740 ;sspi 1:0.19 24 032D00 ;sspi 1:0.19 24 040218 ;sspi 1:0.19 24 040340
;sspi 1:0.19 24 050502 ;sspi 1:0.19 24 050603 ;sspi 1:0.19 24 05070C ;sspi
1:0.19 24 000F01 ;sspi 1:0.19 24 020301 ;sspi 1:0.19 24 000F01 ;sspi 1:0.19
24 020300 ;sspi 1:0.19 24 000F01 ;sspi 1:0.19 24 032A01 ;sspi 1:0.19 24 000F01
;sspi 1:0.19 24 032A00 ;sspi 1:0.19 24 000F01'
> setenv linux_mmc 'run mrcm_init;faload mmc 0 $(loadaddr) vmlinux.64;bootoctlinux
$(loadaddr)'
> saveenv
```

Boot:

```
> run linux_mmc
```

You should see the kernel log messages followed by the terminal prompt (no login required).

Run the Application

On the Network

1. Serve the application and configuration file on an NFS share. Mount the share as follows.
> `mount -o nolock,hard,intr <server ip addr>:/path/to/application /home`
2. Run the application:
> `cd /home`
> `./idtPtp`

On the SD Card

1. Serve the application and configuration file on the SD card. Mount the partition:
> `mount /dev/mmcb1k0p1 /home`
2. Run the application:
> `cd /home`
> `./idtPtp`

NFS Mount

The following instructions show how to set up an NFS server/client on Debian based systems. Execute all commands as root.

Development Host

1. Install the necessary packages:
#`apt-get install nfs-kernel-server`
#`apt-get install nfs-common`
#`apt-get install rpcbind`
2. Create the directory to be NFS mounted as shown in the following example:
#`mkdir /mnt/cavium`
#`chmod 777 /mnt/cavium`
3. Edit `/etc/exports` to add the lines similar to the following example:
`/mnt/cavium*(rw,no_root_squash,async)`
4. Export the share and (re)start the NFS server:
#`exportfs -a`
#`service nfs-kernel-server start`

Target Board

Create a local NFS client directory to mount substituting the NFS server IP address in the following example:

```
#mkdir /home  
#chmod 777 /home  
#mount -o nolock,intr,hard <nfs server ip address>:/mnt/cavium /home
```

Revision History

Revision Date	Description of Change
November 1, 2017	Initial release.



Corporate Headquarters
6024 Silver Creek Valley Road
San Jose, CA 95138 USA
www.IDT.com

Sales
1-800-345-7015 or 408-284-8200
Fax: 408-284-2775
www.IDT.com/go/sales

Tech Support
www.IDT.com/go/support

DISCLAIMER Integrated Device Technology, Inc. (IDT) and its affiliated companies (herein referred to as "IDT") reserve the right to modify the products and/or specifications described herein at any time, without notice, at IDT's sole discretion. Performance specifications and operating parameters of the described products are determined in an independent state and are not guaranteed to perform the same way when installed in customer products. The information contained herein is provided without representation or warranty of any kind, whether express or implied, including, but not limited to, the suitability of IDT's products for any particular purpose, an implied warranty of merchantability, or non-infringement of the intellectual property rights of others. This document is presented only as a guide and does not convey any license under intellectual property rights of IDT or any third parties.

IDT's products are not intended for use in applications involving extreme environmental conditions or in life support systems or similar devices where the failure or malfunction of an IDT product can be reasonably expected to significantly affect the health or safety of users. Anyone using an IDT product in such a manner does so at their own risk, absent an express, written agreement by IDT.

Integrated Device Technology, IDT and the IDT logo are trademarks or registered trademarks of IDT and its subsidiaries in the United States and other countries. Other trademarks used herein are the property of IDT or their respective third party owners. For datasheet type definitions and a glossary of common terms, visit www.idt.com/go/glossary. Integrated Device Technology, Inc.. All rights reserved.