Cyclone V SoC Power Optimization

2015.02.09

AN-734 Subscribe Send Feedback

"How do I squeeze another 10m W out of my design?" A question that seemingly only plagued mobile handset designers a decade ago is now an important consideration for nearly every embedded designer. Choosing an Altera® SoC FPGA is the first major step in the direction of minimizing power, saving up to 30% or more compared to previous generation 2-chip solutions (processor/DSP + FPGA). Optimizing the design of the power supply itself can also be of major benefit. However, there are additional steps you can take in optimizing the power consumption in a Cyclone®V SoC. This application note discusses methods to minimize power consumption to help meet the power targets for your design.

Related Information

SoC Development Kit resource page

Power Reduction

Power reduction on SoC FPGAs can be broken up into methods focused on the Hard Processor System (HPS) and FPGA portions of the device.

Refer to the appendix for power measurement techniques and instructions on how to set-up your Cyclone V SoC development kit.

Related Information

- **HPS Power Reduction Methods** on page 1
- **FPGA Power Reduction Methods** on page 6
- **Appendix: Power Measurement Techniques for Cyclone V SoC Dev Kit** on page 12

HPS Power Reduction Methods

The primary methods of power reduction for the HPS are contained in the ARM® Cortex™-A9 microprocessor unit (MPU) subsystem and in the power consumed by peripherals and interfaces.

MPU

The Cortex-A9 MPU subsystem consists of:

- MPCore
  - CPU0
  - CPU1
  - ACP
  - SCU
- L2 Cache
- ACP ID Mapper

Related Information

Cortex-A9 Microprocessor Unit Subsystem
For more information on the MPU subsystem, refer to the *Cortex-A9 Microprocessor Unit Subsystem* chapter in the *Cyclone V Hard Processor System Technical Reference Manual, Volume 3* .

HPS Method 1: Fit Code in Cache

Accesses to and from the DDR SDRAM can consume large amounts of power (>=400 mW). Ensuring that your code fits into the 512KB of the L2 Cache can significantly reduce power consumption requirements. The amount of power saved depends upon the interface type (DDR2, DDR3, and LPDDR2) and speed of the interface.

HPS Method 2: Processor Standby Modes and Dynamic Clock Gating

The Cortex-A9 processor can utilize clock-gating logic throughout the MPU subsystem.

Check that the following are enabled/set prior to placing cores in a clock-gated mode:

1. Enable dynamic clock gating in the processor cores. Use a CP15 (co-processor) assembly command to access the "Power Control Register".

   ```
   asm volatile("mcr p15,0,%0,c15,c0,0"::"r"(0x701));
   ```

   Note: The code is GCC specific. Similar syntax can be used with other compliers.
2. Ensure the IRQ controller and SCU can take advantage of clock-gated mode. Both IRQ and SCU are set/controlled in the SCU control register:
   a. Bit 6 disables the IRQ Controller clock when no IRQs are pending
   b. Bit 5 puts the SCU into standby mode when all processors are clock-gated

   ```
   volatile uint32 *scucr = (volatile uint32*) 0xFFFEC000;
   *scucr |= 0x60;
   ```
3. Enable L2 clock-gating and standby mode. In the Power Control register of the L2-310 Cache Controller, ensure that dynamic clock gating and standby mode are both enabled:
   a. Dynamic Clock Gating: Set bit1 of `pwr_ctrl` register
   b. Standby Mode: Set bit0 of `pwr_ctrl` register

   ```
   volatile uint32 *reg15_power_control = (volatile uint32*) 0xFFFEFF80;
   reg15_power_control |= 0x03;
   ```

Please refer to the *AMBA Level 2 Cache Controller Technical Reference Manual* and *Cortex-A9 MPCore Technical Reference Manual* for detailed descriptions of the CP15 register, Snoop Control Unit's (SCU's) control register, and the cache controller's power control register.

After ensuring that dynamic clock gating is used within the MPU subsystem, you can put one or both of the Cortex A-9 processor CPUs into clock gating mode by issuing a WFI (wait for interrupt) or WFE (wait for event) assembly instructions. Which one you choose depends on the required system behavior.

Bare-metal design examples illustrating the usage of WFI/WFE and the standby mode (dynamic clock gating) that it enables are included in a companion ZIP file (Power_Optimization.zip). Unzip this file to see two examples. In one of the these designs, one CPU is configured to be in and out of standby mode using the SEV instruction. The other design uses pushbutton IRQs to wake the CPU from the WFI mode.

Note: None of these clock-gating modes activate until one of the CPUs initiates the standby mode with a WFI or WFE call. The wake-up trigger from WFI or WFE mode is well described in ARM's Cortex-A9 documentation. Wake up a core in standby with a running core by issuing a SEV instruction.

Related Information

- **AMBA Level 2 Cache Controller (L2C-310) Technical Reference Manual**
- **Cortex-A9 MPCore Technical Reference Manual**
- **Cortex-A9 Processor Power Control**
  For more information on standby modes, refer to the *Cortex-A9 Technical Reference Manual* under the *Cortex-A9 Processor Power Control* section.
- **SoC Design Example**
  For the power optimization example code files.

HPS Method 3: CPU1 in WFI/WFE or Standby Loop

On boot, CPU1 is held in reset. If you do not intend to make use of the CPU1, you should put it into a Standby Loop in order to benefit from around 90 mW of power savings.

1. Place the following code for CPU1 at a known address in on-chip memory:

```
while(1)
{
    __asm__("WFE");
}
```

   Note: The code is GCC specific. Similar syntax can be used with other compliers.
2. Configure this address in the `cpu1startaddr` register in the system manager.
3. Pull CPU1 out of reset (the default) using the cpu1 register in the reset manager.

Related Information

SoC HPS Address Map and Register Descriptions
For more information on register level details, refer to the SoC HPS Address Map and Register Descriptions web page.

HPS Method 4: Lowering CPU Frequency

Lowering CPU frequency can reduce the amount of power the MPU consumes as shown in the following table:

Table 1: Power Reduction for Different CPU Frequencies

Frequency Change		Power Savings
From:	To:	Approximate Power Reduction:
800 MHz	400 MHz	190 mW
400 MHz	200 MHz	90 mW
200 MHz	100 MHz	50 mW

The Rocketboards Preloader Clocking Customization link indicates, changing SDRAM frequency is a separate exercise and should only be done using Qsys.

Note: Be aware that changes to the PLL VCOs can affect certain peripheral speeds. On the Preloader Clocking Customization link you can use the value generating tables to check if the desired peripheral speeds are met before making changes and recompiling the Preloader.

Related Information

Preloader Clocking Customization

To change the frequency, follow the instructions and use the web tool on Rocketboards under Preloader Clocking Customization.

Peripherals

In addition to the MPU subsystem, the Cyclone V SoC HPS contains a collection of hardened peripherals and interfaces:

- 2 x 10/100/1000 Ethernet MACs
- 2 x USB 2.0 OTG Controllers
- 2 x UARTs
- 2 x SPI Masters
- 2 x SPI Slaves
- 4 x I^2C

The majority of the power consumed by the HPS portion of the SoC device is in the MPU and the SDRAM interface. Two additional peripherals that deserve further consideration when it comes to power savings are USB and Ethernet.

HPS Method 5: Energy Efficient Ethernet

Ethernet MAC's in the HPS support Energy Efficient Ethernet (EEE).

Refer to the "Programming Guidelines for Energy Efficient Ethernet" section in the *Ethernet Media Access Controller* chapter of the Cyclone V technical reference manual for detailed information on how to take advantage of this power saving standard for Ethernet.

Both sides of an Ethernet connection must support EEE in order to take full advantage of it. The Ethernet drivers in the current version of Linux for SoC fully support EEE.

Related Information

Ethernet Media Access Controller

For more information on the Ethernet Media Access Controller, refer to the *Ethernet Media Access Controller* chapter in the *Cyclone V Hard Processor System Technical Reference Manual, Volume 3*.

HPS Method 6: USB Power Management

The USB controllers in the HPS support USB standard methods of power reduction. Like Ethernet, these modes cannot be used unless the host and device support the desired power reduction method.

Refer to the Linux kernel documentation (`./Documentation/usb/power-management.txt`) for more information on USB power management. Points to highlight from the power management documentation:

- `CONFIG_PM_RUNTIME` must be set
- Use sysfs files to control/test device and system power features
  - With `CONFIG_PM_RUNTIME` enabled, a mouse returns a similar list like the one below in the `./power` sub-directory for the device in sysfs:

```
root@socfpga:~#ls/sys/bus/usb/devices/1-1.2/power/

active_duration              wakeup
autosuspend                  wakeup_abort_count
autosuspend_delay_ms         wakeup_active
connected_duration           wakeup_active_count
control                      wakeup_count
level                        wakeup_expire_count
persist                      wakeup_last_time_ms
runtime_active_time          wakeup_max_time_ms
runtime_status               wakeup_prevent_sleep_time_ms
runtime_suspended_time       wakeup_total_time_ms
```

HPS Method 7: Unused Peripherals in Reset

You should ensure that the reset manager holds each unused peripheral in reset. Refer to the `rstmgr` section in the SoC HPS Address and Register Descriptions website. Qsys and the Preloader control the initial reset state for each peripheral, you can reset individual peripherals under software control via the reset manager. Unlike the core (MPU) logic, the peripherals are designed to be most power efficient when held in reset.

Related Information

SoC HPS Address Map and Register Descriptions

For more information on register level details, refer to the SoC HPS Address Map and Register Descriptions web page.

HPS Power Reduction Summary

There are several methods for reducing HPS power consumption.

1. Fit code in cache
   - Ensure that the loop you're running (code and data being accessed) doesn't consume more than the 512Kb of L2 cache.
2. Processor standby modes and clock gating

- Follow the instructions preceding the WFI/WFE call to take advantage of dynamic clock gating in the MPU
- Issue WFI or WFE instructions to enter standby mode
  - Entering this mode engages any dynamic clock gating logic you've enabled

3. Put CPU1 in WFI/WFE or "standby loop" when CPU1 is not be used in your system.
4. Reduce the overall CPU clock frequency
5. Utilize low power modes in Ethernet
6. Utilize low power modes in the USB controller
7. Hold unused peripherals in reset

The table below gives an order of magnitude estimate of the potential power savings that can be gained using each of these methods:

Table 2: Order of Magnitude Power Savings for HPS Power Reduction Methods

HPS Power Saving Method	Approximate Power Reduction
Code Loop in L2 Cache	400 mW
CPU Frequency Reduction	~0.475 mW/MHz
WFI/WFE or Standby Loop	200 mW
WFI/WFE vs CPU1 held in reset	90 mW(1)
Ethernet EEE/Idle	20-30 mW
USB Idle	10-15 mW

Note: These numbers are approximate and are provided to give visibility to relative power reduction gains for the listed methods. They are provided to help prioritize power saving efforts for the biggest payoff. They are not guaranteed and will vary from system to system or SoC to SoC. All numbers with the exception of the "Frequency Reduction" are based on an 800 MHz CPU clock.

Related Information

SoC Design Example

For the power optimization example code files.

FPGA Power Reduction Methods

The Cyclone V FPGA power consumption reduction techniques are well documented on Altera's website and in numerous academic and industry documents. Any FPGA power reduction method that applies to the Cyclone V FPGA also applies to the FPGA portion of the Cyclone V SoC as they both use the same FPGA fabric. Additionally, the Cyclone V SoC offers is the ability to power down the FPGA portion of the device while the HPS remains running. This feature is explained in this section.

FPGA Power Consumption

Because the elements of the HPS are hardened, they generally consume less space and therefore less power than the FPGA portion of the SoC device. Both static and operational power tend to run higher on the FPGA portion than on the HPS portion of the device.

(1) It is best to put an unused core in a WFE/WFI loop to take advantage of the power reduction.

For example, the Altera PowerPlay Early Power Estimator (EPE) was used to examine a high power demand case where a 110KLE Cyclone V SoC ran a Linpack benchmark test on both CPUs in the Cortex-A9 processor operating at 800 MHz with heavily loaded FPGA logic. In this example, 69% of the total power was dynamic power consumed by the FPGA logic, 23% was HPS dynamic plus static power, and the remaining 8% is FPGA static power. Therefore, it is advantageous to be able to power down the FPGA side of the device to remove as much of the FPGA dynamic power as possible when it is not being used.

Related Information

- **Cyclone IV and Cyclone V PowerPlay Early Power Estimator**
- **SoC Design Example**
  For the power optimization example code files.

FPGA Portion Power Down

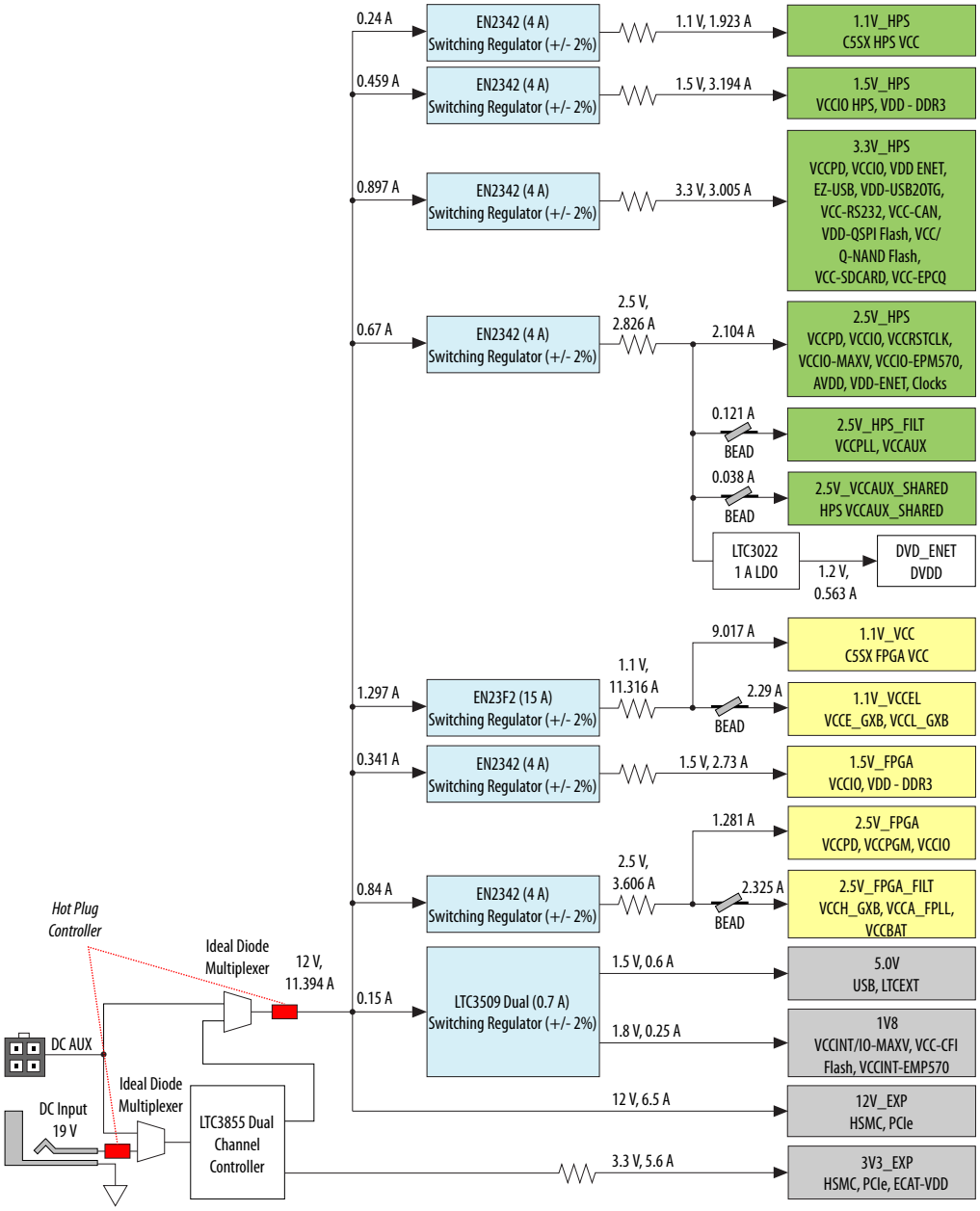
Altera SoCs have separate power planes for the HPS and FPGA portions of the device. Due to separate power planes, the Cyclone V SoC offers the ability to power down the FPGA when it's not being utilized while the HPS portion remains running. To implement the FPGA power down mode, the board power rails must be designed properly as an initial foundation. Additionally, the FPGA must be inactive with the HPS-to-FPGA bus interfaces held in reset before the FPGA portion is powered down. The FPGA power down is triggered by executing a power off command. As with any device power down mode, there will be some wake-up time to bring the FPGA portion back to life. This time is based on the FPGA configuration time.

FPGA Power Off Step 1: Board Design (Power Rail) Choices

The Cyclone V SoC supports combining power rails for the HPS and FPGA. To implement FPGA Power Down Mode, be sure to provide separate power sources for all rails that can be shared between the HPS and FPGA circuitry. Good references to follow are available as part of the Cyclone V SoC Development Kit download, which can be found on the *Cyclone V SoC Development Kit and SoC Embedded Design Suite* web page.

Altera recommends using Altera's Enpirion® Power Solutions for the required Cyclone V SoC power rails. Altera provides several resources to assist with understanding your Cyclone V SoC power requirements as well as selecting components for and designing a suitable power tree. These resources can be found at Altera's Powering FPGAs Resource Center web page.

Figure 1: Cyclone V SoC Development Kit Reference Power Supply Design



Related Information

- Powering FPGAs Resource Page
- Enpirion Power
- SoC Development Kit resource page

FPGA Power Off Step 2: Quiet FPGA

The FPGA must be inactive and connectivity between the HPS and FPGA must be held in reset when a power or configuration/re-configuration event is initiated. These conditions are similar to those you would

have when re-configuring or power cycling a discrete FPGA. Ensuring that your design is not issuing transactions over any of the HPS-to-FPGA bridges is the minimum requirement.

One way to do this would be to implement a register-controlled signal in your FPGA design logic that would end all activity and perform an ordered shutdown. Once the ordered shutdown is complete, an ACK bit could be set so the last transaction would be reading that bit. The last transaction would likely be over the `lwhps2fpga` bridge.

Once your FPGA design is quiet, you must ensure that the bridges between the FPGA and HPS side of the SoC device are disabled. Do this by putting all the bridges in reset. Visit the SoC HPS Address and Register Descriptions page for the Cyclone V and find the `brgmodrst` register, located at address 0xFFD0501C in the reset manager register section.

Below is how to put each bridge register in reset from within u-boot connected to a Cyclone V SoC Development Kit:

```
SOCFPGA_CYCLONE5 # mm 0xffd0501c

ffd0501c: 00000000 ? 7

ffd05020: 00000000 ? q
```

Using u-boot's `mm` command and the information on the SoC HPS Address and Register Descriptions page mentioned above, the bridges are placed in reset. By setting a value of 7 (0'b0111) to the `brgmodrst` register, each bridge is held in reset and it is now safe to power cycle the FPGA side of the SoC.

Related Information

SoC HPS Address Map and Register Descriptions

For more information on register level details, refer to the SoC HPS Address Map and Register Descriptions web page.

FPGA Power Off Step 3: Power Off the FPGA

Because there are no sequencing requirements for powering off the FPGA portion of the SoC device, issue the power off command that is needed to trigger the power management for your system to power off the FPGA portion.

For instance, it's common to make use of the `CONTROL0` or `CONTROL1` pin on the LTC2978 device to control power on/off in a system. Routing either or both of these pins to a GPIO on the HPS side of the SoC device and then writing a 0 or 1 to the pin can power cycle the FPGA portion.

FPGA Power Off Step 4: Wake up Event for Power on and FPGA Configuration

There are two stages to restoring FPGA functionality:

- FPGA power up
- FPGA configuration

For typical designs, writing a 1 to the GPIO that is connected to CONTROL0 (or CONTROL1) triggers power up. However, with the LTPowerPlay probe connected, this operation cannot function due to the removal of the I^2C resistors. With the probe connected, power must be applied through the probe by individually powering up each rail. FPGA configuration/re-configuration is necessary after power-up. The FPGA image can be pulled from many options such as the EPCQ device or loaded from the HPS side.

Refer to the "Requirements" section of this app note for information regarding the removal of the I^2C resistors.

Related Information

Requirements on page 16

FPGA Power Off Step 5: Power On and FPGA Reconfiguration Time Considerations

The two important time periods to consider when reapplying power to the FPGA portion of the SoC device are the power on time (system/power management specific) and the FPGA configuration time.

Note: "Power on time" refers to how long it takes for the FPGA power rails to reach nominal voltage levels.

Of these two, configuration time dominates and is dependent on the mode of configuration selected. From the programming examples shipped as part of the SoCEDS software (`$SOCEDS_DEST_ROOT/examples/software/Altera-SoCFPGA-HardwareLib-FPGA-CV-GNU.tar.gz`), the following steps are used to program the FPGA:

1. The "power on" bit (in the FPGA manager's status register) is checked to see if the FPGA is on.
   - If off, configuration fails.
   - If on, the HPS takes control of the configuration block.
2. The MSEL (configuration mode bits) are read.

   Note: Only the modes listed in the *FPGA Manager* chapter in the Cyclone V reference manual are supported.
3. If the MSEL are set to a supported mode, a DMA is configured to transfer data from a buffer into the FPGA configuration block.
   - Data is written (4 bytes at a time) into the configuration block.
4. On success, the FPGA completes configuration and enters user mode.

Given that the binary files used to configure the FPGA (**.rbf**: raw binary files) are in the range of 7 MB, the following table estimates configuration times:

Table 3: FPGA Configuration Times based on Configuration Type

Configuration Type	Approximate Time (seconds)
HPS Configuration	0.15-0.2
Active Serial (via external EPCQ)	1.0

These configuration times assume **.sof/.rbf** is resident in SDRAM.

For more information about the configuration process, refer to the *FPGA Manager* chapter in the *Cyclone V Hard Processor System Technical Reference Manual, Volume 3.*

Related Information

Cyclone V Hard Processor System Technical Reference Manual

Power Optimization Summary

The Cyclone V SoC offers significant power reduction potential through the mere fact of integration. This power savings benefit can be further extended through use of the power saving methods described in this application note to help meet the power targets for your design. On the processor side, seven methods are available. As for the FPGA, powering off the FPGA when not utilized provides the most dramatic

difference. Whatever your system power savings goals are, the methods described in this application note can help make a difference in achieving them.

Appendix: Power Measurement Techniques for Cyclone V SoC Dev Kit

Power Monitoring and Measurement

Before beginning to optimize power consumption, it is critical that accurate measurements of power can be taken on the board. There are three different methods for measuring and monitoring power on the Cyclone V SoC Development Kit.

- Altera's Power Monitor application
- Linear Technology's LTpowerPlay™ software
- Linux drivers for monitoring the power from within Linux running on the on-board Cortex-A9 processor.

Note: Have the latest version of Java installed before proceeding with any Linux instructions.

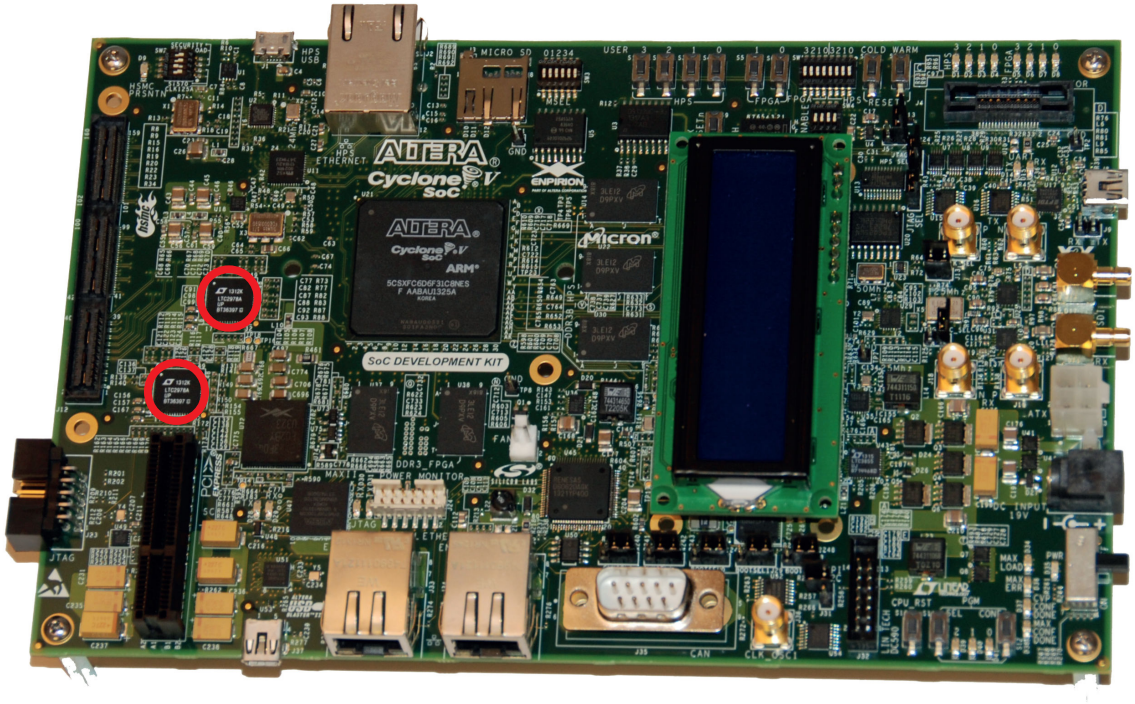
Related Information

- **Cyclone V SoC Development Kit Power Monitor Application** on page 13
- **LTC LTpower Play Tool** on page 15
- **Using the LTC2978A Linux Driver** on page 17

Cyclone V SoC Development Kit Power Management ICs

The Cyclone V SoC Development Kit uses several power SoC DC-DC step-down converters from Altera's Enpirion series. Refer to the *Cyclone V SoC Development Kit Reference Power Supply Design* figure in the *FPGA Power Off Step 1: Board Design (Power Rail) Choices* section for more detail on the Enpirion chips used on the board. The Cyclone V SoC Development Kit also uses two Linear Technology Corporation LTC2978A Power System Manager ICs, see them circled in red in the image below.

Figure 2: Cyclone V SoC Development Kit with LTC2978A Power Management ICs



One of the LTC2978A chips is used for power monitoring and control of the HPS and the other for power monitoring and control of the FPGA.

Note: For all new designs the LTC2977 is recommended.

The on-chip EEPROMs are preprogrammed to properly bring up both the HPS and FPGA sides of the SoC device. An I^2C/PMBus/SMBus slave allows a processor or external application to control and monitor system power supplies. Both the Cyclone V SoC Development Kit Power Monitor application and the LTpowerPlay tool make use of this I^2C bus to monitor temperature, power consumption, and to control the various power rails.

Related Information

- **FPGA Power Off Step 1: Board Design (Power Rail) Choices** on page 7
- **Enpirion Power**
- **Linear Technology Corporation: LTC2978A**
  For more information on the LTC2978A and LTC2977, please contact Linear Technology Corporation.

Cyclone V SoC Development Kit Power Monitor Application

The power monitor application that is shipped with the Cyclone V Development Kit reads the output voltage for each power rail on the kit.

Starting the Application

For Windows or Linux:

1. Browse to the Cyclone V SoC Development resource page and download the development kit installation package that is appropriate for you.

   Note: Be sure to pay attention to the version to download (ES or production silicon).
2. Get a ZIP package for Linux or self-installing EXE for Windows.

Related Information

SoC Development Kit resource page

Windows Instructions

1. Run the installation executable.
2. Browse the `<Kit install directory>/example/board_test_system`.
   - In this directory locate the executable for the Power Monitor (**PowerMonitor.exe**).
   - Run this executable with the FPGA programmed with one of the designs included with the kit.

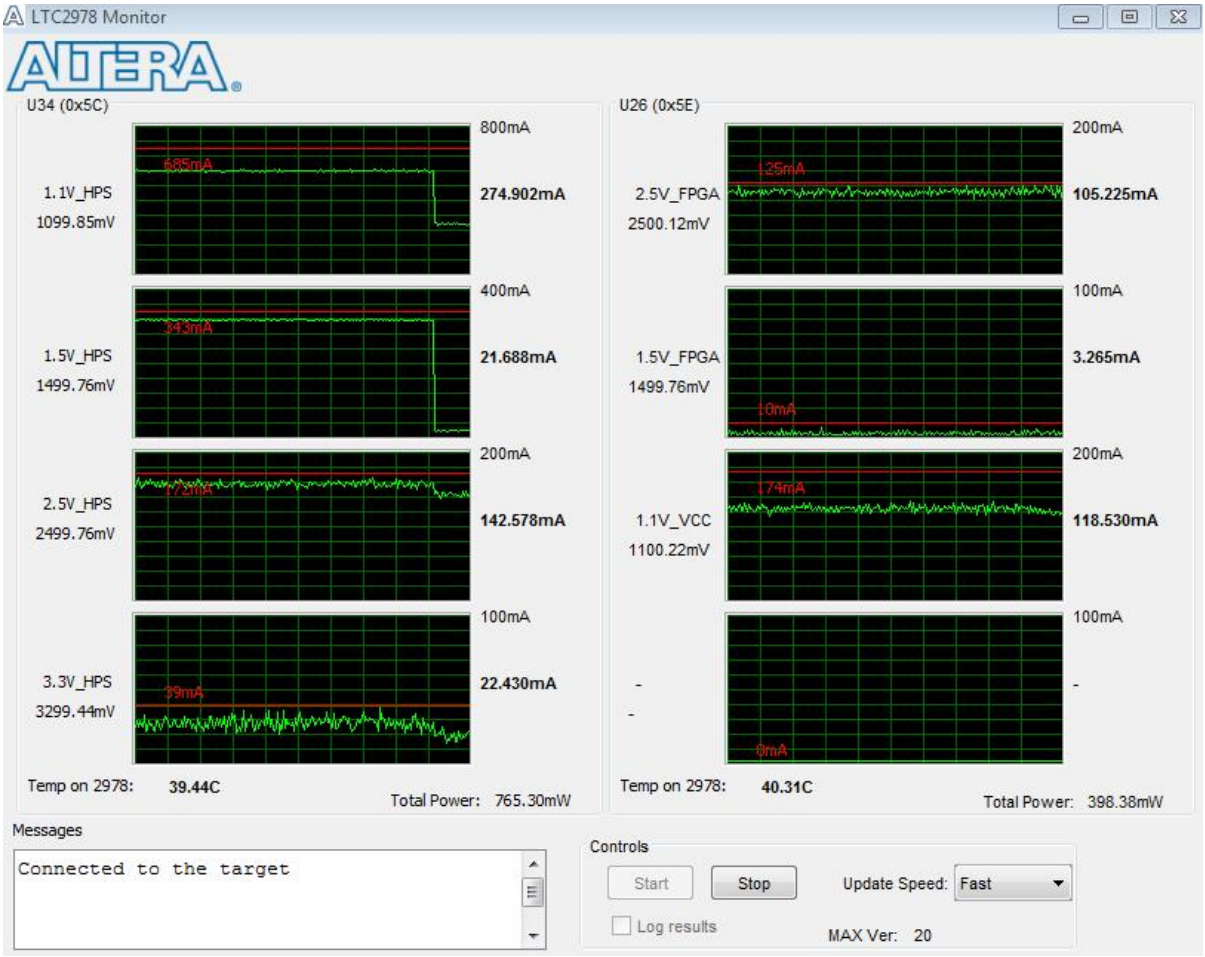
Linux Instructions

1. Unzip to a directory with the following command:
   - `unzip -d <directory><name of ZIP>`

   Be sure to explicitly create the top level directory "<directory>" as the development kit contents do not have a top level directory as part of the archive.
2. Change to the `<Kit install directory>/examples/board_test_system` sub-directory. Run "java -jar pt.jar".

   Note: In order for the Power Monitor to function, the FPGA must be configured with a compatible design. Refer to the User Guide that is part of the kit installation for further detail.

Figure 3: Altera Power Management Application Main User Interface



3. The HPS rails and the power consumption are on the left. The FPGA rails are on the right. Click **Start** in the Controls group to start sampling power at three speeds: slow, medium, and fast.

Related Information

Cyclone V SoC Development Kit User Guide

LTC LTpower Play Tool

In addition to the LTC2978A power management ICs, Linear Technology also offers the LTpowerPlay GUI software that you can use to help you monitor power consumption. The LTpowerPlay software requires use of the LTC DC1613A controller.

Related Information

Linear Technology contact page

Contact your local Linear Technology representative to receive an LTC DC1613A chip.

Requirements

- LTC LTPowerPlay Tool.
- LTC DC1613A USB to I^2C controller.
- Board must have a DC1613A compatible connector. Refer to the DC1613A Compatible Connector figure.

Note: Minor board re-work may be required. It's likely that the board you're using has zero ohm resistors that prevent the USB from being a master on the I^2C bus for communicating with the LTC2978A devices.

Figure 4: DC1613A Compatible Connector

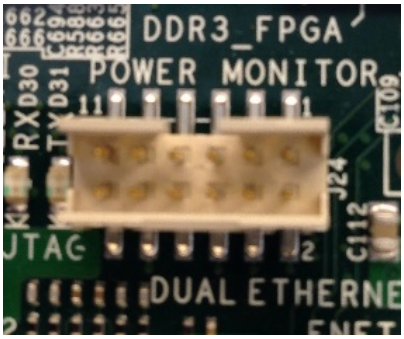


Figure 5: Schematic Representation of 0 Ohm HPS and FPGA Resistors

Figure 6: Microscope View of Resistors

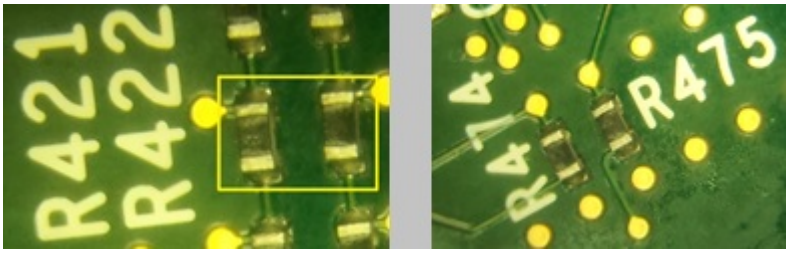
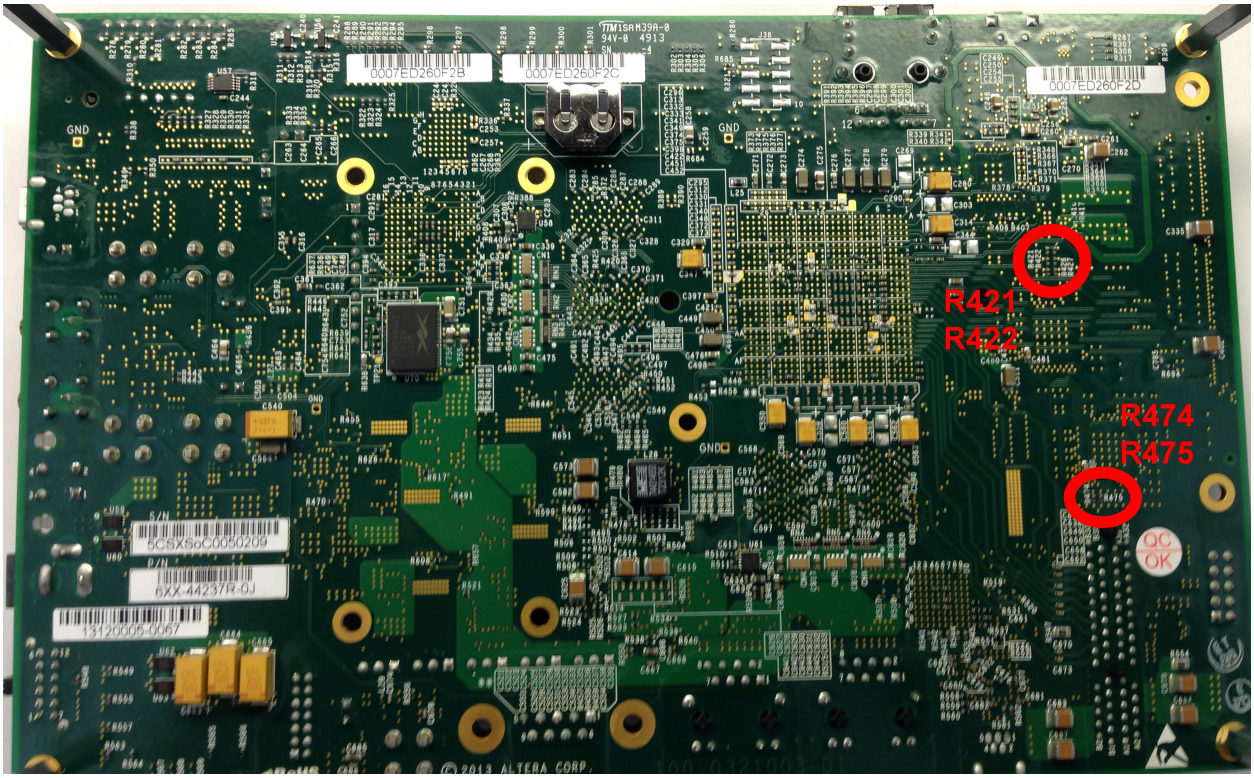


Figure 7: Relative Position of Resistors



Once these resistors are removed, the on-board I^2C connectivity to the LTC power management ICs is disabled, meaning the HPS can no longer read temperature from LTC2978As, etc. These aspects must now be read through the LTpowerPlay GUI.

Related Information

- **LTpowerPlay Tool Download**
  To download the Linear Technology LTpowerPlay Tool, refer to the LTpowerPlay web page.
- **DC1613A**
  For more information on the LTC DC1613A USB to I^2C controller, refer to the DC1613A Linear Technology web page.

LTC Video Tutorial

The following video provides an excellent introduction for usage flow using the LTPowerPlay tool on the Cyclone V SoC Development kit.

Related Information

Digital Power System Management on the Altera Cyclone V SoC Development Board Video

Using the LTC2978A Linux Driver

A third approach to monitoring Cyclone V SoC Development Kit power is accomplished using the LTC2978A Linux driver. The Linux kernel provides hardware monitoring support for the LTC2978A device.

The following sections are a quick description of how to add the driver to Linux and make use of it. Further information can be found in (`./Documentation/hwmon/ltc2978`).

For the testing done for this application note, kernels 3.12 and 3.10 were used. The instructions on Rocketboards for building a kernel from GIT provided the guidelines to follow while completing the steps to add the driver to Linux.

Related Information

Getting Started Using Git Trees

Compiling the Driver

After setting up your environment to include a path to the Linaro ARM GCC compiler as is mentioned in the "Linux Kernel" instructions, follow the instructions until after you've completed "make `ARCH`=ARM socfpga_defconfig" step. This step creates a kernel with the default configuration for the "socfpga" (Altera's SoC FPGA) devices.

1. Run "make `ARCH`=ARM menuconfig" and select:

   ```
   Device Drivers --->
   ```

   Followed by **ENTER**.
2. In the "Device Drivers" section look for:

   ```
   <> Hardware Monitoring support --->
   ```

   Note: Use the space bar to toggle the <> from disabled <> to enabled <*>. Setting it to <M> adds it as a module, which was not tested.

   Hit **ENTER** to enter go into the "Hardware Monitoring support" section. Browse to: `<*> PMBus support --->`
3. Set the <> to <*> and hit **ENTER** to go to the PMBus support section.

   In this section, find <> Linear Technologies LTC2974, LTC2978, LTC3880, and LTC3883.
4. Change the <> to <*> to enable the LTC2978A Hardware Monitoring Driver.
5. Follow the instructions on Rocketboards to build the kernel.
6. Make ARCH=arm uImage `LOADADDR`=0x8000.
7. Copy the resulting zImage from `./arch/arm/boot` to the FAT partition of your SD Card and boot Linux.

Using the Driver

1. Boot the Linux just compiled. From the prompt, add a "new device" to the "sysfs" representation of data from LTC2978A family devices.

   Command format: echo ltc2978 <i2c addr> > /sys/bus/i2c/devices/i2c-<n>/new_device

   Where "i2c addr" is the address of the device on the I^2C/PMBus bus and "<n>" is the number of the bus where it resides. Standard I^2C utilities can be used to probe the buses.
2. Add the two LTC2978A devices, using the following commands:
   - HPS Power Monitor: echo ltc2978 0x5c > /sys/bus/i2c/devices/i2c-0/new_device
   - FPGA Power Monitor: echo ltc2978 0x5e > /sys/bus/i2c/devices/i2c-0/new_device
3. Once the devices are added successfully, files are created in the "sysfs" filesystem for input and output voltages, temperatures and more. Please refer to the LTC2978A datasheet and the kernel documentation mentioned in the "HPS Method: USB Power Management" section for further information on what this driver has provided and how to make use of the data.

Alternately, you can use raw I^2C commands to get at the same information, although this driver does some of the number conversions for you.

Related Information

- **LTC2978 Dtatsheet**
- **HPS Method 6: USB Power Management** on page 5

Power Measurement Results on Cyclone V SoC Development Kit

Once you have your preferred power measurement set configured and running, you can take power measurements of the Cyclone V SoC device. The table below shows results across four different boards in four different lab locations.

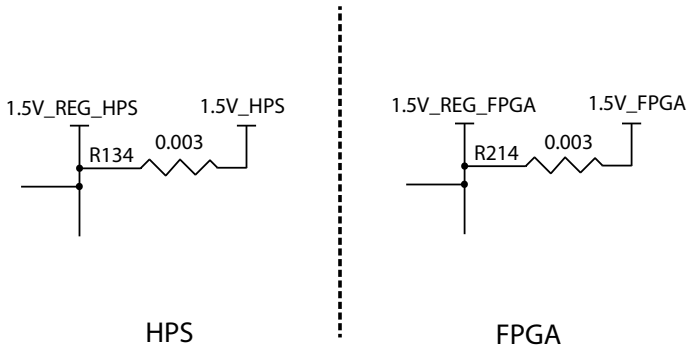
Table 4: Linux Idle Power Measurement on Cyclone V SoC Development Kit

Power Rail (V)	Board 1 Rev A Dev Kit Measured (W)	Board 2 Rev C Dev Kit Measured (W)	Board 3 Rev C Dev Kit Measured (W)	Board 4 Rev C Dev Kit Measured (W)	Average (W)
HPS 1.1	0.390	0.40	0.39	0.39	0.393
HPS 1.5	0.081	0.08	0.08	0.08	0.080
HPS 2.5	0.488	0.62	0.47	0.59	0.542
HPS 3.3	0.106	0.06	0.09	0.08	0.084
FPGA 2.5	0.039	0.03	0.05	0.02	0.035
FPGA 1.5	0.007	0.003	0.007	0.01	0.007
FPGA 1.1	0.059	0.13	0.08	0.15	0.105
Total Power	1.170	1.32	1.17	1.31	1.246

Measurements were taken at room temperature running Linux idle. The FPGA portion of the SoC is programmed with the Golden Hardware Reference Design (GHRD) which is included standard with the development kit.

Because a (3mΩ) sense resistor is used on both the HPS 1.5V and the FPGA 1.5V rails, measurements from the board should be divided by 3 on those rails. The results above show the divided by 3 values.

Figure 8: Sense Resistors



The board revision can be found on the back of the Cyclone V SoC Development Kit board with Rev A being the initial engineering sample version. The latest production version have upgraded to the Enpirion power supply solution.

The results are fairly consistent across the different boards with some minor variance in different rails. Once your equipment is set up, these results can be used to verify your measurement techniques are being done properly.

Related Information

Power Reference Designs
Save time with Power-Optimized FPGA Reference Designs.

Document Revision History

Table 5: Document Revision History

Date	Version	Changes
Feburary 2015	2015.02.09	Initial release.