EZ-Color Device Capability and Selection Guide

AN47217

Authors: Mukund Krishna and Ben Kropf

Associated Project: No
Associated Part Family: CY8CLED04, CY8CLED08, CY8CLED16
GET FREE SAMPLES HERE
Software Version: PSoC Designer™ 4.4 or PSoC Express™ 3.0
Associated Application Notes: None

Application Note Abstract

EZ-Color™ controllers are optimized for High Brightness LED (HB LED) control applications. This application note describes the capabilities of the EZ-Color family of controllers and details resource usage with generic examples. The high configurability and varied uses of these controllers makes this application note pertinent.

Introduction

This application note presents the capabilities of the EZ-Color family of HB LED controllers. It provides an overview of the device resource considerations to be made when choosing the right EZ-Color device for an application. Common functions that EZ-Color devices implement are:

- LED Dimming Modulation
  - Pulse density modulation techniques
- Digital Communication for Lighting
  - DMX512
  - DALI
- LED Temperature Compensation
- CapSense
- 3 and 4 Channel Color Mixing
  - Including LED binning compensation
- Optical Feedback Algorithms

LED Dimming Modulation

The LED dimming modulators are an important part of any HB LED application. All EZ-Color controllers are capable of three primary types of LED dimming modulations. These are:

- Pulse Width Modulation (PWM)
- Precise Intensity Stochastic Modulation (PrISM™)
- Delta-Sigma Modulated PWM (DSPWM)

Pulse Width Modulation is among the most commonly used and conventional methods of modulation. It is straightforward to use and effective in practice. There are two additional techniques of modulation that are superior to using the PWM alone supported by EZ-Color.

PrISM is a modulation technique that is developed and patented by Cypress. It results in reduced EMI as compared to the PWM technique while still providing adequate dimming control for LEDs.

The Delta Sigma Modulated PWM technique provides higher resolution while utilizing equal hardware resources as a conventional PWM.

LED dimming modulators use digital block resources. Digital blocks are configurable 8-bit digital peripherals. For all devices in the family, there are two types of digital blocks—basic and communication. Usually, there are equal numbers of each. Any communication functions must be implemented using communication blocks but basic, non-communication functions are implemented using either kind of block.

Table 1. Resources Available for EZ-Color Controllers

Device	Digital Basic/Comm Blocks	Analog CT/SC Blocks	Flash (kB)	RAM (B)	CapSense	USB
CY8CLED04	2/2	2/4	16	1024	√	√
CY8CLED08	4/4	4/8	16	256		
CY8CLED16	8/8	4/8	32	2048		

Table 1 lists the number of digital blocks available for each EZ-Color controller in the family. PWM and DSPWM modulators can have a dimming resolution of up to 16 bits. A PrISM modulator can theoretically have a dimming resolution of up to 32 bits, but the maximum recommended resolution for these modulators is 13 bits. This is because the output signal of a PrISM modulator has a frequency output range that increases with the resolution of the modulator. This increase in frequency output range is undesirable as it goes beyond the switching frequency of the current driver. Therefore, a resolution of 13 bits or lower is recommended for a PrISM modulator. Refer to AN47372, PrISM Technology for LED Dimming. To determine the number of digital blocks used by one PWM or PrISM modulator, use Equation 1. Note that a partial digital block cannot be used, so the result must always be rounded up. In Equation 1, *n* is the dimming resolution of the modulator. The resolution of dimming is determined by the color accuracy needed for the end application.

$$DigBlocks_{PWM,PRISM} = \frac{n}{8} \qquad \text{Equation 1}$$

Equations 2 and 3 are used to determine how many digital blocks are needed by a DSPWM. The total dimming resolution of a DSPWM modulator is the total of the hardware PWM modulation resolution and extra resolution added by delta-sigma modulation in the software. Equation 3 shows that the number of digital blocks needed is only determined by the hardware resolution.

$$n_{Total} = n_{SW} + n_{HW} \qquad \text{Equation 2}$$

$$DigBlocks_{DSPWM} = \frac{n_{HW}}{8} \qquad \text{Equation 3}$$

From these equations it can be determined that more dimming resolution is achieved with a DSPWM modulator than with a PWM or PrISM modulator. The tradeoff is that a DSPWM modulator requires more code space and execution time to use.

Equations 1, 2, and 3 determine the number of digital blocks required by one modulator. The total number of blocks for all modulators is determined by adding up the digital blocks needed by each modulator used in the device.

It is clear that any EZ-Color device has a variety of LED dimming configurations. For example, the CY8CLED16 device has 16 digital blocks. Therefore it can implement eight 16-bit PWM modulators, eight 12-bit PrISM modulators, or sixteen 12-bit DSPWM modulators (assuming the software resolution is 4 bits). As another example, it can implement four 10-bit PrISM modulators and still have eight digital blocks left over to implement other digital functions.

Digital Communication

Most HB LED based lighting systems require some form of digital communication to send or receive data to or from the light fixtures to control them. Table 2 lists digital communication protocols that can be implemented with hardware on EZ-Color controllers. Some of the hardware is dedicated for a protocol and does not use any digital blocks. Some protocols use digital blocks to implement the communication.

A DMX512 protocol receiver can be implemented using two digital blocks. This is a standard protocol that is common in stage and concert lighting systems. The receiver has software programmable address and programmable number of channels that it can control. A typical DMX512 receiver implementation (developed by Cypress) controlling three LED channels consumes five digital blocks (three for the LED modulators). The firmware uses approximately 1K of Flash and less than 20 bytes of SRAM.

Figure 1 is a pictorial representation of the resource usage in a CY8CLED08 for a DMX512 receiver controlling three LED channels. The portions shaded green represent resources or parts of them that are used for the design. As expected, some resources such as clocks, interconnects, and the processor are used in any design.

Table 2. Digital Communication Resource Use

Data Protocol	Digital Blocks	Comm Digital Blocks
DMX512 (Receiver)	2	1
DALI (Slave)	3	0
I^{2}C™ Master or Slave	0	0
Full-Duplex UART	2	2
Half-Duplex UART (or Single RX/TX)	1	1
USB*	0	0
SPI Master or Slave	1	1

*Available only for CY8CLED04

DALI is another lighting communication protocol that is common for large commercial buildings. The DALI slave can be implemented in EZ-Color consuming six digital blocks (3 for the DALI slave and 3 to modulate 3 LED channels). The three blocks used to implement DALI need not be communication blocks as the Manchester encoding is performed in the software. The IP developed by Cypress uses approximately 16K of Flash and 220 bytes of SRAM. If it is implemented without automatic addressing, the Flash usage is reduced by approximately 3K.

Apart from these specific lighting communication protocols, the industry standard communication protocols such as I^{2}C, UART, and SPI can be implemented in any of the devices in the family. As examples, SPI can be used to interface to external WUSB devices while I^{2}C can be used to interface to 'PowerLine' modems. Additionally, the CY8CLED04 also has a USB interface.

Table 2 also shows the number of digital block resources that each type of communication block consumes.

As a reference, a simple 8-bit receiver consumes only 350 bytes of incremental Flash and 20 more bytes of SRAM. A UART implementation consumes approximately 500 bytes of Flash and at least 3 bytes of SRAM. More SRAM usage depends on the sizes of the buffers.

An SPI master or slave consumes about 30 to 50 bytes of Flash on its own.

For I^{2}C, the Flash memory usage depends on the implementation. A simple I^{2}C slave can consume only 300 bytes of Flash but a big multi-master slave implementation can consume up to 2K of Flash.

There is a dedicated USB interface on the CY8CLED04 device that does not utilize any of the digital block resources.

LED Temperature Compensation

Many HB LED systems require the measurement of analog signals. Often one or more thermistors are present to measure temperatures of the system and the LEDs. Any EZ-Color device that measures an analog signal does so with an Analog-to-Digital Converter (ADC). The EZ-Color device is capable of implementing a wide range of flexible ADC implementations. The ADCs cover a wide range of resolutions and techniques and use varied number of digital and analog block resources. For help in selecting from this multitude of ADCs, refer to the application note AN2239, *Analog – ADC Selection* on www.cypress.com. When designing with an EZ-Color device, the number of digital and analog blocks used by an ADC implementation must be factored into the total number of digital and analog blocks that are used.

In a typical case, such as the 3-channel color mixing firmware IP developed by Cypress, the simple 8-bit incremental ADC is used. This module occupies one digital and one switched capacitor analog block.

Analog blocks come in two flavors, continuous time and switched capacitor blocks. The former enables continuous time functions such as comparators and Programmable Gain Amplifiers. The switched capacitor blocks enable functions such as ADCs and filters.

As a reference for the amount of Flash used, the 3-channel color mix firmware with temperature compensation occupies approximately 6K of Flash when designed using PSoC Designer™ 4.4. The incremental amount of Flash memory used for the temperature compensation component of the firmware is approximately 1.5K using PSoC Designer.

Temperature sensors with an I^{2}C interface can also be used instead of raw thermistors, thereby eliminating the need for ADCs and complicated processing.

CapSense

The CY8CLED04 device has the resources to implement the Cypress capacitive touch sensing technique known as CapSense™. The primary resource that makes this effective is an analog multiplexor bus that allows any GPIO pin to be an input to the analog system of the device. This allows many capacitive copper pads to be used to create CapSense buttons, sliders, and touchscreens. If CapSense technology is required in an HB LED application, digital block resource use must be carefully considered. Implementing CapSense requires one, two, or three digital blocks. The CY8CLED04 device has four total digital blocks. Therefore it is likely that the CapSense implementation that only uses one digital block must be used. This allows three free digital blocks for other functions such as LED dimming. The CY8CLED04 can interface to a total of 46 sensors. This includes buttons and sliders. If the diplexing method is used, then more sensors can be interfaced (only for slider applications). Refer to Cypress application notes, AN2394, *CapSense Best Practices* and AN2292, *Capacitance Sensing – Layout Guidelines for PSoC CapSense* for further details on CapSense implementation.

Color Mixing Algorithm

Code algorithms to implement color mixing functionality work well with EZ-Color controllers. Color mixing algorithms convert a set of color coordinates that specify a color into the appropriate 8-bit dimming values for the LED dimming modulators. This allows the EZ-Color controller to be communicated with on a higher level and maintain desired color and brightness levels.

The basic 3-channel color mixing firmware performing 8-bit LED dimming requires three 8-bit dimming blocks. From the discussion on LED Dimming Modulation, it can be inferred that it consumes three digital blocks. The addition of a simple temperature compensation algorithm using a thermistor consumes an additional digital block and analog block (for the ADC).

If the dimming resolution is increased, the number of digital blocks needed should be calculated accordingly.

The 3-channel color mix firmware with temperature compensation developed at Cypress using PSoC Designer 4.4, consumes approximately 6K of Flash memory in the device and about 4.5K without the temperature compensation. In either case, the SRAM usage does not exceed 100 bytes.

Cypress' PSoC Express™ software (a code free, high level design tool) includes a driver that implements a color mixing algorithm, temperature feedback, and LED dimming modulators in one building block. This driver includes many functionalities and therefore, uses a large amount of Flash space (about 12K) for code. Figure 2 shows a pictorial representation of the resource usage for a basic 3-channel color mix design with temperature compensation.

ColorLock Algorithm

ColorLock functionality uses feedback from an optical sensor in the system to adjust the LED dimming modulators correctly to “lock on” to a target color. This is similar to the concept of temperature compensation because—it compensates for change in color. Instead of indirectly measuring change in color through temperature, it senses actual change in color and compensates for it.

The ColorLock algorithm implemented by Cypress requires the use of 10 digital blocks. Due to a 9-bit PrISM implementation, six digital blocks are used for dimming as per Equation 1. A 16-bit PWM and two 8-bit timers are also used that form the frame generator, pulse counter, and de-bounce counter.

The implementation using PSoC Designer consumes approximately 16K of Flash. Using the PSoC Express driver to implement the colorlock algorithm to control a 3-channel color mixing solution with optical feedback consumes slightly over 16K of Flash.

Other Functions

EZ-Color devices are also capable of many functions other than those discussed earlier in this application note. Most functions that can be implemented with a standard microcontroller can be also implemented with the right EZ-Color device. For instance, an application may require Li-Ion battery charging in addition to LED dimming. EZ-Color devices are able to realize other functions such as battery charging.

Similar to regular PSoC devices, EZ-Color controllers also have the Dynamic Reconfiguration ability. This is a technique that allows the reuse of the device’s digital and analog resources for different functions that need not be available simultaneously. For instance, an LED multicolor flashlight may be powered by Li-Ion batteries and require battery charging functionality as part of the controller. The LED dimming and the battery charging do not necessarily need to happen at the same time. Therefore, the digital and analog blocks that implement the LED dimming functionality can dynamically reconfigure into resources that implement battery charging. By doing this, an EZ-Color device gets more functionality out of a fixed number of resources than would otherwise be possible. The only constraint on this technique is the amount of Flash and SRAM size required for the code to implement these functions.

Summary

The EZ-Color devices are a versatile and powerful family that is suited to developing innovative and differentiating designs for LED lighting applications.

The choice of device is the result of a careful analysis of the application, leading to a breakup of the resources required. Based on this detail and the resources on each device, the optimum device is chosen.

Figure 3 can be used as a starting point to estimate the resources needed for an application.

Figure 1. Resource Usage of a DMX512 Receiver Controlling 3 Channels

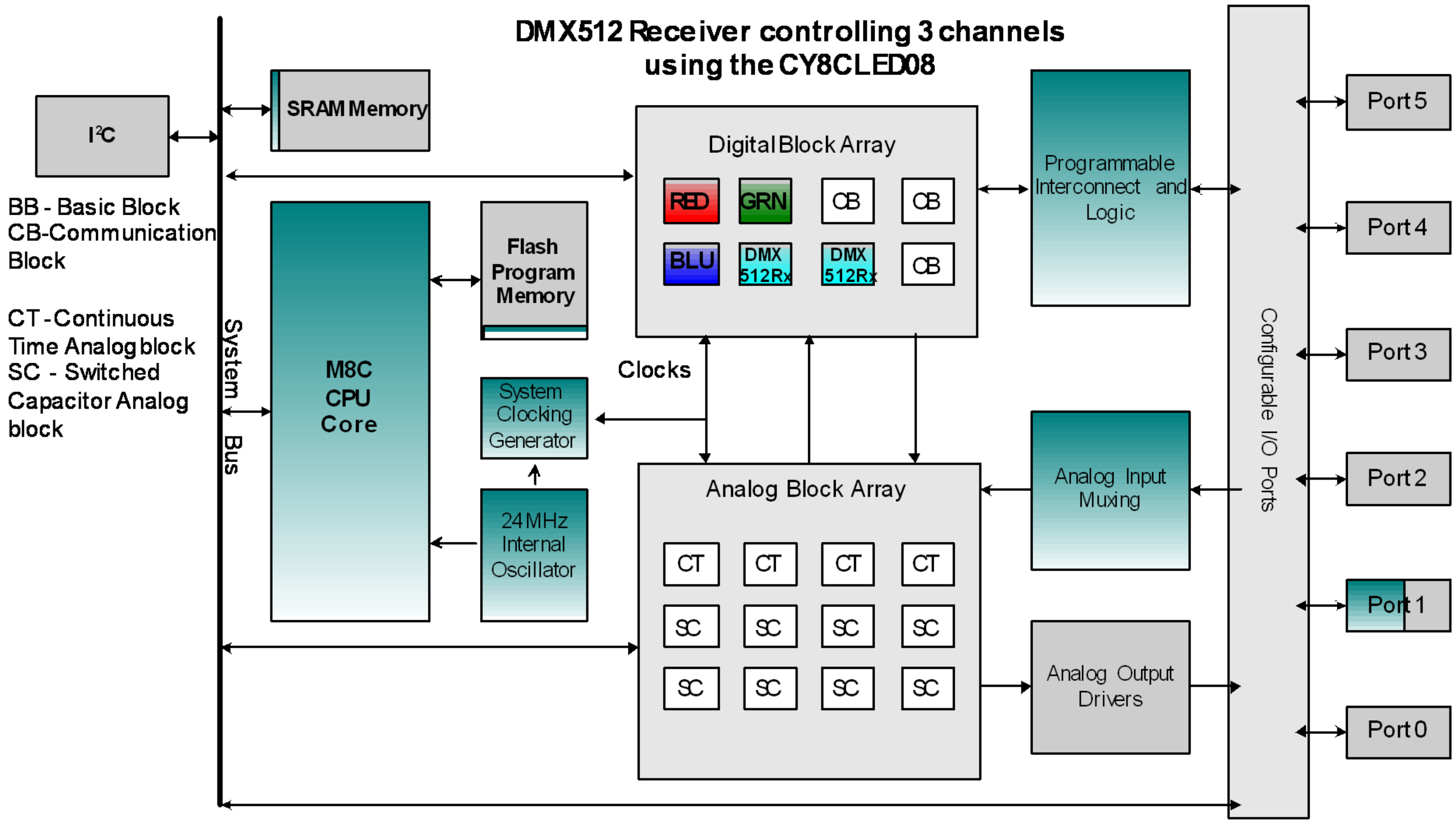


Figure 2. Resource Usage of 3 Channel Color Mix Firmware with Temperature Compensation

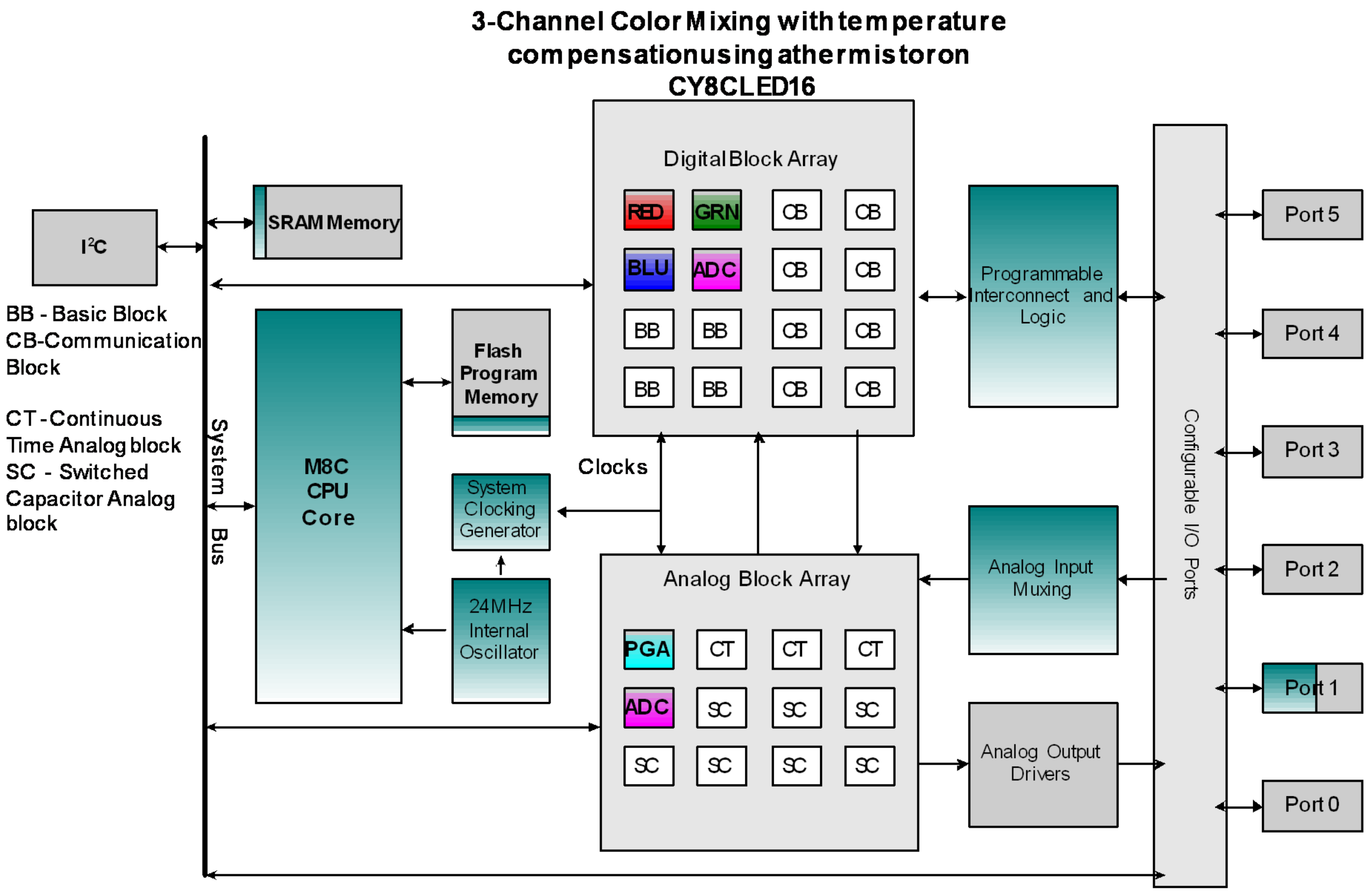
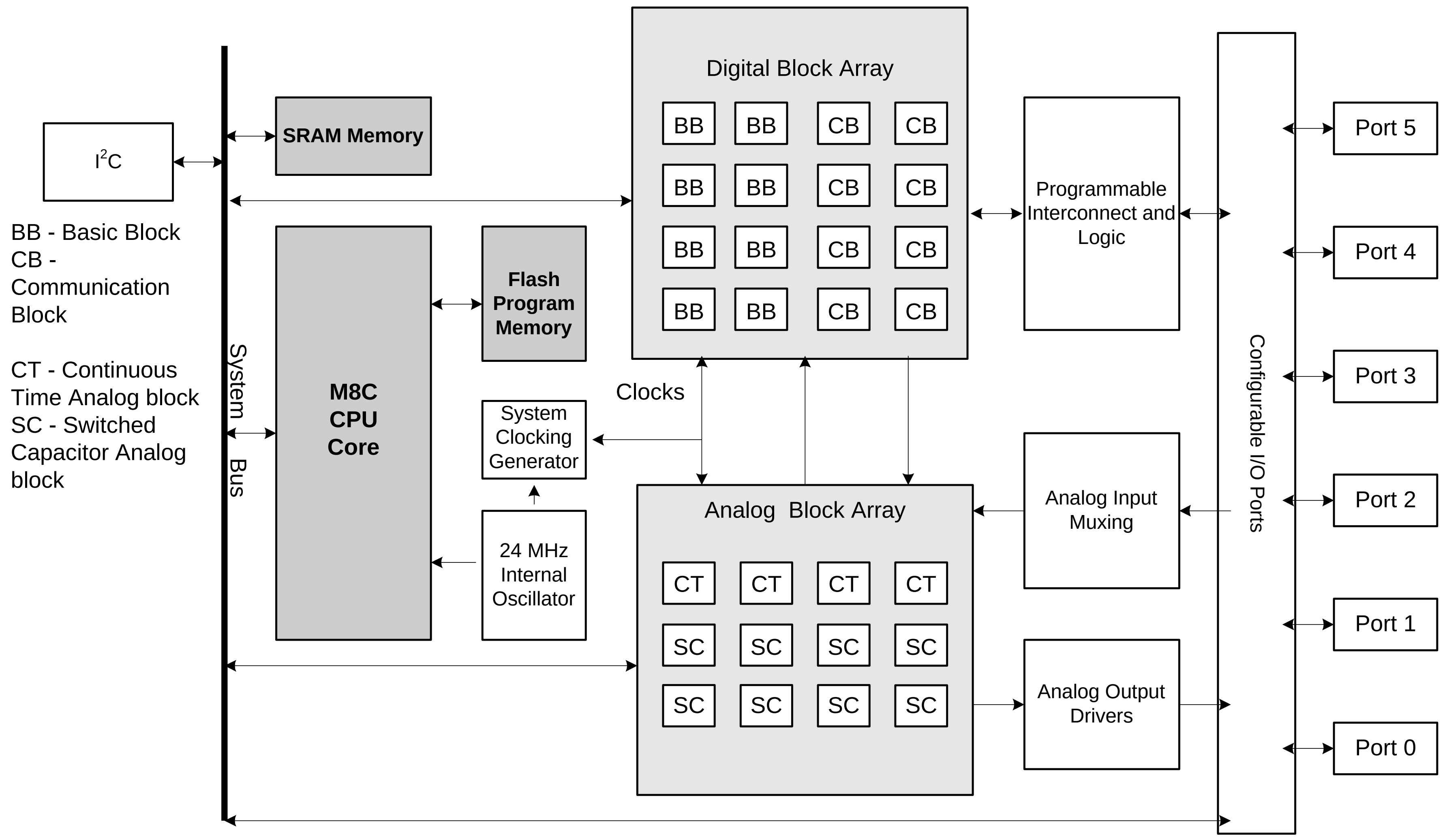


Figure 3. Use this Page to Approximate the Resource Usage for your Design

CY8CLED16



BB - Basic Block
CB - Communication Block

CT - Continuous Time Analog block
SC - Switched Capacitor Analog block

About the Authors

Name:	Mukund Krishna
Title:	Applications Engineer
Background:	Mukund has a graduate degree in Electrical Engineering from the University of Southern California. He currently works in the lighting solutions group, where he is responsible for aiding customer designs, designing collateral, and product definitions.
Contact:	Email: Mukund.Krishna@cypress.com
	Ph: (408)-432-7058

Name:	Ben Kropf
Title:	Applications Engineer
Background:	Ben graduated with a B.S in Electrical Engineering from Seattle Pacific University. His professional experience includes mixed-signal embedded systems design, firmware design, and high brightness LED applications.
Contact:	Email: btk@cypress.com

Document History Page

Document Title: EZ-Color Device Capability and Selection Guide

Document Number: 001-47217

Revision	ECN	Submission Date	Orig. of Change	Description of Change
**	2529400	07/08/08	UKK/BTK	New Application Note

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com/