

Whitepaper

Beyond Price: How to Assess the Real Value of an 8-bit MCU

Beyond Price: How to Assess the Real Value of an 8-bit MCU

By Mark Beecham, MCU and Sensor Product Manager, Silicon Labs

Introduction

Today's 8-bit MCUs still hold a large portion of the overall microcontroller (MCU) market, despite competition and strong market share gains from newer platforms like Arm Cortex-M processors. The continuing strength of 8-bit MCUs in the embedded market rests on multiple factors: low cost, simplicity, quick control capabilities and mature ecosystems with a ubiquitous tool base.

As 8-bit has matured as an MCU technology, the price per unit has approached its lower limits, creating a hidden challenge for developers. When looking at low-cost 8-bit MCUs for an embedded design, there are many good options with similar prices and feature sets, but which is the best? One aspect often overlooked is what is the true price and value of similar 8-bit MCU hardware from different vendors. Here are a few important considerations that will ultimately reveal the real price of an 8-bit MCU.

Ecosystems

Tools

When cost is a driving factor of an embedded design, then it's wise to avoid spending capital on compiler licenses or coding environments. Before choosing the MCU, confirm that the vendor and the software examples they supply are available on a free development platform. Otherwise, depending on component volume, the true cost of one vendor may be lower than another even if the average selling price (ASP) doesn't paint the same picture.

For example, two 8051-based MCU options from competing vendors may have similar hardware specs but different tool costs. If an MCU vendor lacks a compiler or integrated development environment (IDE) license for their 8051-based devices, the developer must use Keil or IAR and pay the license fees, increasing the total investment for the project. A more cost-effective option is to use an 8-bit platform supported by a free IDE and unlimited Keil license.

Software

Once the compiler and development environment are understood, the next hidden specification is the availability of software examples and the ecosystem on the MCU platform. For example, look for MCU vendors offering a plethora of code examples for their peripherals. This makes it easy to grab each peripheral needed such as a PWM, UART and ADC, then combine them into one project and finish the design sooner. This can mean getting to market faster and potentially increasing your revenue, which makes the typically higher ASP of a well-supported MCU ecosystem worth the price premium.

Contents

- Introduction
- Ecosystems
 - Tools
 - Software
- Specifications
 - Guaranteed by Design, Characterization and Test (GBD, GBC, and GBT)
- Flexibility and Scalability
- Longevity and Supply Assurance
- Conclusion

Specifications

Guaranteed by Design, Characterization and Test (GBD, GBC and GBT)

When perusing specifications of relatively simple 8-bit MCUs, it is easy to understand the capabilities of the device from the summary page in the datasheet and then reference the electrical specification tables for more details on the important specs. However, it is a bit more complicated than it appears, and developers should consider three key aspects: 1) whether there are minimum and maximum values specified for what is important to the design, 2) what the test conditions of those values will be compared to the actual use case, and 3) whether these values are guaranteed by design, characterization or test.

Typical values always need to be considered with caution as temperature, Vdd level, operating frequency and other factors can affect what the value truly will be in a design. An all-too-common occurrence in the industry these days is experiencing limited functionality in one spec based on another spec. This is typically done to achieve a very attractive spec on the front page of a datasheet. However, after digging into the electrical specification tables, it becomes obvious that the seemingly topnotch specification only exists under very limited parameters such as Vdd, core frequency, temperature, etc., which may clash with other aspects of the design. This illusion can lead to disappointment as at first the best solution jumps out on the front page, but then the design decision becomes less clear after discovering all the caveats of those specifications.

For example, the graph shown in Figure 1 comes from the front page of a datasheet. The 20 MHz spec is great for such an inexpensive MCU; however, the fine print indicates this is only achievable above 4.5V Vdd, which might not be possible in a system or incur the cost of using a larger boost converter to achieve that operating speed.

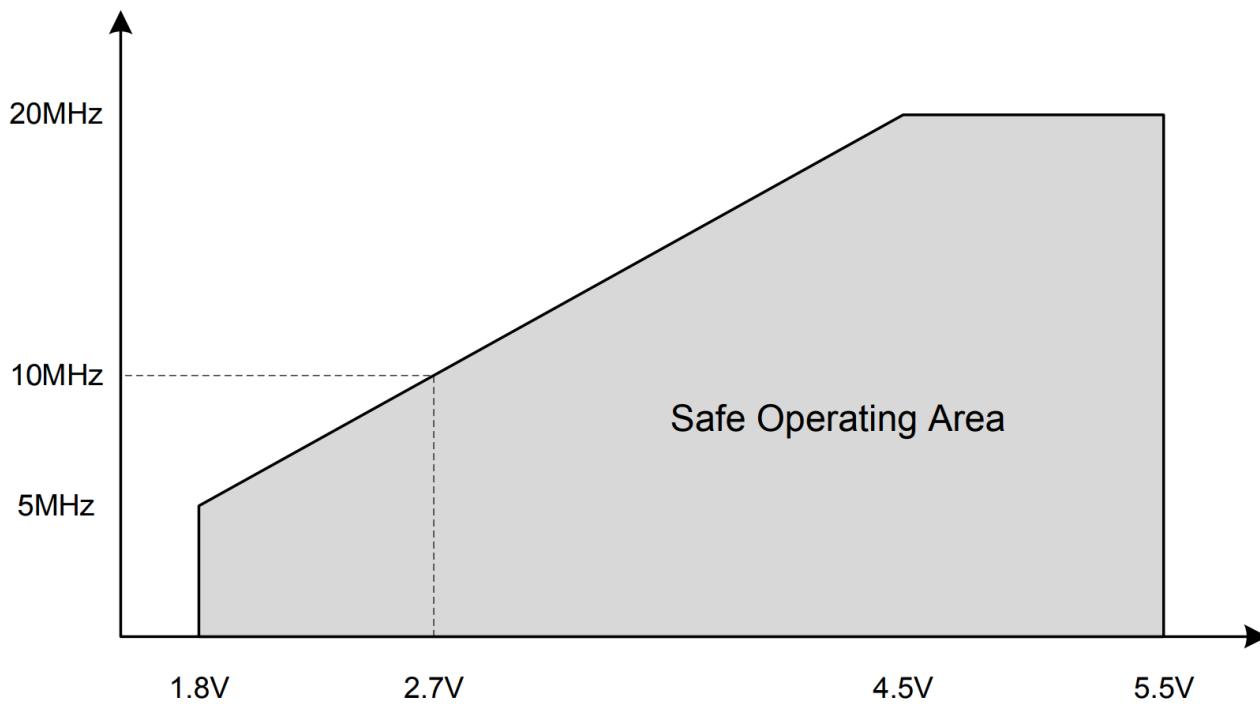


Figure 1. Graph of safe operating areas.

The graph in Figure 1 raises questions about other specs depending heavily on other factors. When an important specification has a test condition of 4 MHz, but the design will operate at 20 MHz, the spec needs to be taken with forethought that it might not be accurate for certain use cases. One can assume it could be drastically inaccurate, especially if it's an analog specification.

Additionally, when examining specification tables, it's important to study the footnotes and understand if it is guaranteed by design, characterization or a test referred to as GBD, GBC and GBT respectively. Typically, GBD offers the lowest confidence level in the spec, while GBT instills the most trust in the specification with GBC in the middle.

The example in Table 1 (from a typical MCU product datasheet) shows two options of GBD and GBC. The GBD is a bit worrisome if the application has strict timing requirements and requires a very accurate oscillator as the only spec given for the trimmed high-speed oscillator is GBD. This could cause a high-speed communication interface like UART to fail if it approaches the five percent inaccuracy number of the untrimmed high-speed internal (HSI) oscillator. In applications such as instrumentation or metering, where you need to count or keep track of events in a certain time window, a drift in the oscillator will affect the accuracy of what is being measured.

Table 1. HSI oscillator characteristics for a typical 8-bit MCU.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{HSI}	Frequency	-	-	16	-	MHz
ACC _{HSI}	Accuracy of HSI oscillator	User-trimmed with the CLK_HSITRIMR register for given V_{DD} and T_A conditions ⁽¹⁾	-	-	1.0 ⁽²⁾	%
	Accuracy of HSI oscillator (factory calibrated)	$V_{DD} = 5$ V, -40 °C $\leq T_A \leq 85$ °C	-5	-	5	
$t_{su(HSI)}$	HSI oscillator wakeup time including calibration	-	-	-	1.02 ⁽²⁾	μs
$I_{DD(HSI)}$	HSI oscillator power consumption	-	-	170	250 ⁽³⁾	μA

1. See the application note.
2. Guaranteed by design.
3. Data based on characterization results.

Flexibility and Scalability

Jumping between architectures and technologies for each project can delay the completion of the design, slowing time to market. While a device from vendor A may be the best choice for the project at hand, vendor B may have another device that's better for a project coming later in the year. This becomes a balance of optimization for each project, and reuse of development and knowledge between projects. When evaluating a vendor choice for a current design, make sure they have the right solution for future products. For example, a one-off device that's the best option for a project may delay the next project if you need to use a device from a different vendor. Always look for a scalable 8-bit platform, such as Silicon Labs' EFM8 portfolio in Figure 2, which offers scalable memory and GPIO options and is supported by a free IDE and unlimited Keil license. Scalable platforms offer many GPIO and memory options in a similar architecture, allowing easy device switching between projects.

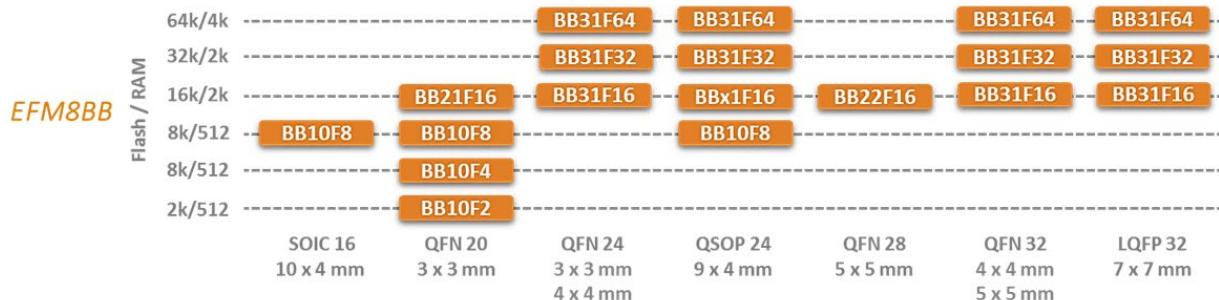


Figure 2. Example of Silicon Labs EFM8 MCU platform specifications.

It also pays to look closely at each device in an MCU family to ensure the features are consistent so that migrating to a larger GPIO device doesn't sacrifice something important like the right number of communication ports, a DAC or a PWM channel.

Longevity and Supply Assurance

The 8-bit market is mature, and 8-bit devices have been around for decades. As a result, 8-bit ASPs are now very low. This is great for developers but can be a pain point for semiconductor vendors as they have approached the floor of profitability, and some vendors have pulled away from making new investments in their 8-bit portfolios. This situation can become alarming when vendors issue end-of-life (EOL) and "not recommended for new design" (NRND) notices, which can threaten the longevity of an end product. Silicon Labs is among a handful of MCU vendors continuing to invest in 8-bit technology. This demonstrates a strong level of commitment to the 8-bit market and instills confidence that a core group of suppliers will not discontinue their MCU products.

Many vendors publicize information on MCU product longevity. Some even offer exact dates of how long the vendor plans to support the devices. For example, Silicon Labs has a minimum date of support listed for each 8-bit family, clearly stating which ones should be used for longer-term design. For a list of minimum longevity dates for Silicon Labs 8-bit products, visit <https://www.silabs.com/products/mcu/8-bit/longevity-commitment>.

MCU product longevity may not be a critical concern for quick designs, such as consumer products with a short lifespan and a quick ramp down, but it is critical for medical, automotive, and industrial applications, where end products take two to three years to design in, then ramp slowly and flatten for potentially more than 10 years. Losing the ability to continue to build an important, profitable end product because an inexpensive 8-bit MCU had been discontinued would be disastrous.

Conclusion

As 8-bit MCUs have carved out their place in the ever-evolving MCU landscape, embedded developers must factor in new considerations. This is essential as the benefits of using an 8-bit MCU include relatively low cost and ease of use, which can be affected by hidden costs of tools, insufficient supporting software, misleading datasheet parameters and lack of scalability. Keeping all of these considerations in mind when assessing the right MCU for your next design can greatly enhance your chances of market success, now and for the long run.

Mark Beecham MCU and Sensor Product Manager, Silicon Labs

Mark Beecham is a product marketing engineer supporting and defining Silicon Labs' 8 and 32-bit microcontroller and sensor products. Specializing in embedded systems and sensors, Mark joined Silicon Labs in 2015 and previously has worked for IBM and Texas Instruments. He holds a BSEE degree from Texas A&M University.