TECHNOLOGIES > MEMORY

Selecting the Correct Memory Type for Embedded Applications

Jan. 4, 2007

By Alex Martinez, Memory Products Division Microchip Tecnologies The type of memory a designer selects for an embedded project drives overall system operation and performance, so obviously this is a very important decision. Whether the

Staff

By Alex Martinez,

Memory Products Division Microchip Tecnologies

The type of memory a designer selects for an embedded project drives overall system operation and performance,
very important decision. Whether the
ries or AC power, the application's
ine whether volatile or nonvolatile
d whether the memory to be used is
ata parameters or both. Additionally,
t play a major role in the selection
ems may have enough memory within
the microcontroller to meet system requirements, while

larger systems may require external memory. A few design parameters to consider when selecting the appropriate memory type include:

- ***Microcontroller selection***
- ***Voltage range***
- ***Battery life***
- ***Read/write speeds***
- ***Memory size***
- ***Memory volatility***
- ***Erase/write endurance levels***
- ***Overall system costs***

Figure 1 illustrates the memory types most commonly used in embedded applications. The following text is a set of guidelines for selecting memory.

GUIDELINES FOR SELECTING MEMORY

1. INTERNAL VS. EXTERNAL MEMORY

Usually, designers make the decision to use internal or external memory after they define the required amounts of code space and data memory. Internal memory is typically the most cost effective memory type, but it is also the least flexible. For this reason, designers must determine future growth possibilities and whether there is an upgrade path to microcontrollers with larger code space. Since cost is

icrocontroller with the least amount

application is typically selected.

be taken when predicting code size,

ode size might require a different

External memory devices are available in various sizes, so it is easy to add memory to accommodate an increase in code size. This sometimes means replacing existing memory with a larger memory device in the same package, or adding additional memory devices to the bus. Even if the microcontroller has internal memory, an external serial EEPROM or Flash device can support requirements for non-volatile memory.

2. BOOT MEMORY

In larger microcontroller or processor-based systems, the design might use boot code for initialization. The application usually determines if boot code is needed and whether a separate boot-memory device is required. For example, if no external address bus or serial boot interface exists, internal memory is generally used, and a separate boot device is not required. However, in designs that do not have internal program memory, initialization is part of the operational code, so all code will reside in the same external program memory. Some microcontrollers have both internal memory and an external address bus. In this case, the boot code could reside in internal memory, and the operational code in external memory. This is probably the safest method, because the operational code can change without accidentally modifying the boot code. In all cases, ...st be non-volatile.

...ON MEMORY

...able Gate Arrays (FPGAs) or System-
...igns, memory devices are used to
...nformation. This memory must be
..., EEPROM or Flash. In most cases,
... some older ones still use an FPGA

serial interface. Serial EEPROM or Flash devices are used most frequently, rather than EPROMs.

4. PROGRAM MEMORY

All systems with a processor utilize program memory, but the designer must determine whether this memory will reside internally or externally. Once this decision is made, the designer can move on to memory density and type. Of course, there are occasions when the microcontroller has both internal program memory and an external address bus, so the designer may choose to use either memory, or a combination of both. This is why selecting the best memory for the application is often complicated by the choice of microcontroller and why changing the memory size can also force a change of microcontrollers.

If the microcontroller utilizes both internal and external memory, internal memory is usually used for code that does not change often, and external memory for code and data that are updated more frequently. A designer also needs to consider whether the memory will be reprogrammed in-circuit or replaced with a new, programmed device. In applications that require reprogrammability, a microcontroller with internal Flash is the typical choice, but a microcontroller with internal OTP or ROM and external Flash or EEPROM also meets this
cost, external Flash can be used for
pace. However, care must be taken to
l modification of code when storing

In most embedded systems, Flash is used for program memory to allow for in-circuit firmware upgrades. Older applications with stable code can still use ROM and OTP memory, but more and more applications are moving toward Flash because of its versatility. Table 1 is a comparison of program memory types.

5. DATA MEMORY

Like program memory, data memory can reside within the microcontroller or in external devices. However, there are a few differences. Data memory such as SRAM (volatile) and EEPROM (non-volatile) are sometimes both available within the same microcontroller, but at other times an internal EEPROM is not available. In these cases, a designer can select an external serial EEPROM or a serial Flash device when large amounts of data are involved. Parallel EEPROMs or Flash are also available, but typically used for program memory only.

When external, high-speed data memory is needed, the typical choice is parallel SRAM, with an external serial EEPROM device to support non-volatile requirements. In some designs, a Flash device is used for program memory, but has a section reserved for data storage. This saves cost, space and provides nonvolatile data memory.

nory requirements, serial EEPROM
^{2}C, SPI or Microwire communication
ash typically uses the SPI bus. In some
s used for its very fast write speeds,
I^{2}C and SPI serial interfaces. Table 2
of data memory types.

6. VOLATILE VS. NON-VOLATILE MEMORY

All memory types can be classified as volatile or non-volatile. Volatile memory loses data when power is removed from the device, while non-volatile memory maintains the data without power. Designers sometimes use a backup battery with volatile memory to make it behave like a non-volatile device, but this can be more

In systems with continuous power, either volatile or non-volatile memory can be used, but the final decision must be based upon the possibility of a power loss. If the information in memory can be recalled from another source when power returns, volatile memory can be used.

Another reason for choosing volatile memory plus a battery is speed. Even though non-volatile memory devices retain data in the absence of power, they have the disadvantage of longer write-cycle times to store a byte, page or sector of data.

7. SERIAL VS. PARALLEL

When the application is defined, the choice of
be one factor in determining whether
nory is used. Microcontrollers
enough internal memory for larger
e cases, external memory is necessary.
dress bus is usually parallel, the
l data memories will also be parallel.

For smaller applications, microcontrollers with internal memory and no external address bus are commonly used. If additional data memory is needed, an external serial memory device is the best choice. And in most cases, this additional external data memory will be non-volatile.

Boot memory can be serial or parallel, depending upon the design. If the microcontroller does not have internal memory, a parallel, non-volatile memory device is a good choice for most applications. For some high-speed applications, though, an external, non-volatile serial-memory device can be used to boot the microcontroller and allow the main code to operate from either internal or external high-speed SRAM.

8. ELECTRICALLY ERASABLE PROM (EEPROM) VS. FLASH

As memory technologies have matured, the line between RAM and ROM has blurred. Today, several memory types combine features of both, such as EEPROM and Flash. These devices read and write like RAM, but maintain their contents without power, like ROM. Both are electrically erasable and programmable, yet each has its own advantages and disadvantages.

From a software viewpoint, standalone EEPROM and Flash
he main difference is that EEPROM
ied byte-by-byte, while Flash devices
rase and either word, page or sector
ed cell locations. The additional steps
lash require the use of SRAM, so
ned on for a longer period of time,
ore battery power. The designer must

also make sure that there is a sufficient amount of SRAM available to make the modifications.

Memory density is another factor in determining whether to select serial EEPROM or Flash. Standalone serial EEPROM devices are available in densities of 128 KB and below, while standalone Flash is available with densities of 32 KB and above.

Applications above 128 KB are possible with serial EEPROM if multiple devices are cascaded together. High erase/write endurance requirements drive an EEPROM choice as a typical serial EEPROM offers one million erase/write cycles. Flash is typically rated at 10,000 cycles, with a few devices offering 100,000 cycles.

Today, most Flash devices are manufactured with a voltage range of 2.7V to 3.6V. Applications within this voltage range can benefit from the lower costs associated with Flash, if byte addressability or high erase/write endurance levels are not required.

9. EEPROM VS. FERROELECTRIC RAM (FRAM)

Design parameters for EEPROM and FRAM are similar, but FRAM tends to have higher read/write endurance
e speeds. Even with these advantages,
tomers select EEPROM over FRAM
and availability. FRAM is expensive,
use lower-cost serial EEPROM unless
s a mandatory system requirement.

10. DYNAMIC RAM (DRAM) VS. STATIC RAM (SRAM)

DRAM and SRAM are both volatile, but SRAM is mainly used for data memory, although both types can be used for both program and data memory. The primary difference between DRAM and SRAM is the lifetime of the data stored. SRAM retains its contents as long as electrical power is present, but DRAM has an extremely short data lifetime—typically about four milliseconds.

DRAM might seem useless in comparison to SRAM; however, a DRAM controller within some microcontrollers makes DRAM behave like SRAM. The DRAM controller periodically refreshes the stored data before it expires, so the memory contents can be stored as long as necessary.

DRAM is typically used for program memory because of its low cost per bit, so applications with huge memory requirements can greatly benefit from DRAM. Its biggest drawback is its slow speed, but computer systems today use high-speed SRAM as cache memory to supplement the much slower DRAM.

Table 3 summarizes the features of each type of memory mentioned in this article. Designers should keep in mind that each type serves a different purpose. Each has its

esses, so sideby-side comparisons are

to use almost any type of memory to

em requirements, the end application

usually the main drivers in this

a combination of memory types can

best suit the application. For example, PDA designs use both volatile and nonvolatile memory for program and data memory. Permanent programs are stored in non-volatile ROM, while programs and data downloaded by the user are stored in battery-backed, volatile DRAM. Whatever memory type or types are selected, designers must carefully consider all design trade-offs before selecting the memory that will be used for the final application.

Alex Martinez is a Product Marketing Manager for the Memory Products Division of Microchip Technology Inc. He can be reached at alex.martinez@microchip.com.

Company: Microchip Technology Inc.

Product URL: Click here for more information of chip design. As current flows through wires, the voltage drops, causing transistors to switch more slowly. The Apris...

Software announced that its Aprisa digital implementation solution is certified by TSMC for its industry-leading N5 and N4...

VOICE YOUR OPINION!

This site requires you to register or login to post a comment.

Email Address*

ded yet. Want to start the conversation?

Electronic Design eNewsletters

SIGN UP

Load More Content

About Us
Contact Us
Advertise
California Do Not Sell
Privacy & Cookie Policy
Terms of Service