AN997
APPLICATION NOTE

M39432: a FLASH+™ Multiple Memory Device

The new FLASH+ technology, from ST, allows Flash memory and EEPROM to be fabricated together on a single die. Significant cost, speed and power consumption advantages are gained over designs that previously used two separate memories, or that previously used software to emulate the effect of having the two types of memory in a single package. This document briefly considers some of these alternative techniques, and highlights the advantages of the FLASH+ technology solution, as used in the M39432 product.

ADVANTAGES AND DISADVANTAGES OF EACH TYPE OF MEMORY

Electronic memory is usually classified under two main headings: *volatile* (losing its contents when the power supply is removed), or *non-volatile* (retaining its contents even when disconnected from an external power supply). Non-volatile memory can be further subdivided into the following technology families:

- ROM (read only, not erasable, mask programmable at time of manufacture)
- EPROM (bytewise electrically programmable, devicewise UV erasable or not erasable)
- Flash memory (bytewise electrically programmable, blockwise electrically erasable)
- EEPROM (bytewise electrically re-programmable, bytewise electrically erasable)

The key advantages and disadvantages of each family are summarized in Table 1. In general, the cell size increases as the inconvenience of erasing is decreased. Since the chip size depends on the cell size, the cost of the memory is proportional to its ease of erasure. Choosing the most appropriate compromise for the application becomes an important design decision within any project.

Table 1. Key Advantages and Disadvantages of Each Family of Non-volatile Memory

Memory Type	Advantages	Disadvantages
ROM	Very small cell size	Cannot be modified except at high cost (a new mask set)
EPROM	Small cell size	The application must be taken out of service for UV erase
FLASH	Medium cell size; electrical erase	Erase only in blocks (sectors)
EEPROM	Bytewise re-programming	Cell size too large for cost effective megabit memories

APPLICATIONS USING A MICROCONTROLLER CHIP AND ONE EXTERNAL MEMORY TYPE

Most microcontroller systems need at least two types of memory:

- a relatively large amount of program memory that does not need to be changed very often
- a small area of scratch memory, for the temporary storage of transient data.

Many microcontroller systems also need an area of non-volatile memory to be set aside for storing persistent data (data that is changed from time to time, but that must be retained from one power-on period to the next). These three areas of memory can be implemented as summarised in the table on the following page.

Table 2. Suitable Memory Technology for Each of the Different Data Types

Data Type	Suitable Technology
Executable program instructions, and large tables of fixed data	ROM, EPROM or Flash memory
Persistent data	Flash memory or EEPROM
Transient data and cache storage	RAM or register banks within the microcontroller chip itself

The demands on all three areas is growing annually. To take a typical example, such as a digital cellular phone, the current requirements, versus those of the near future, are as follows:

- Flash memory, for the executable program
  - currently: 4 Mb (512 Kx8) with a 5 V power supply
  - required soon: 8 Mb (512 Kx16) with a 3 V power supply
- EEPROM, for storing fabrication parameters, user set-up parameters, tables, directories
  - currently: Parallel, 64 Kb with a 5 V power supply
  - required soon: Serial (to reduce package size), 64 Kb with a 3 V power supply
  - There is already a very large market for low capacity, serial EEPROMs, and the market for larger capacity EEPROMs is growing steadily, with a trend to ever greater capacities. When the I^2C™ bus was first developed, 16 Kb was considered adequate as an upper limit for the address space. However, this limit has long been exceeded by applications in robotics, mobile telephones, test equipment and industrial controllers. Microcontroller applications routinely require 256 Kb EEPROMs, and will soon require 1 Mb and even 4 Mb products. This was, of course, anticipated by the introduction of the extended I^2C bus, allowing addressing up to 4 Mb.
- SRAM, for holding transient data, for use as DSP scratch-pad memory and for storing voice messages
  - currently: 64 Kb, or more, with minimum power consumption, and access times of 100 ns
  - required soon: 256 Kb or 1 Mb, or more, with minimum power consumption, and access times of 70 ns

For a more modestly sized application, one possible arrangement is to use just two chips:

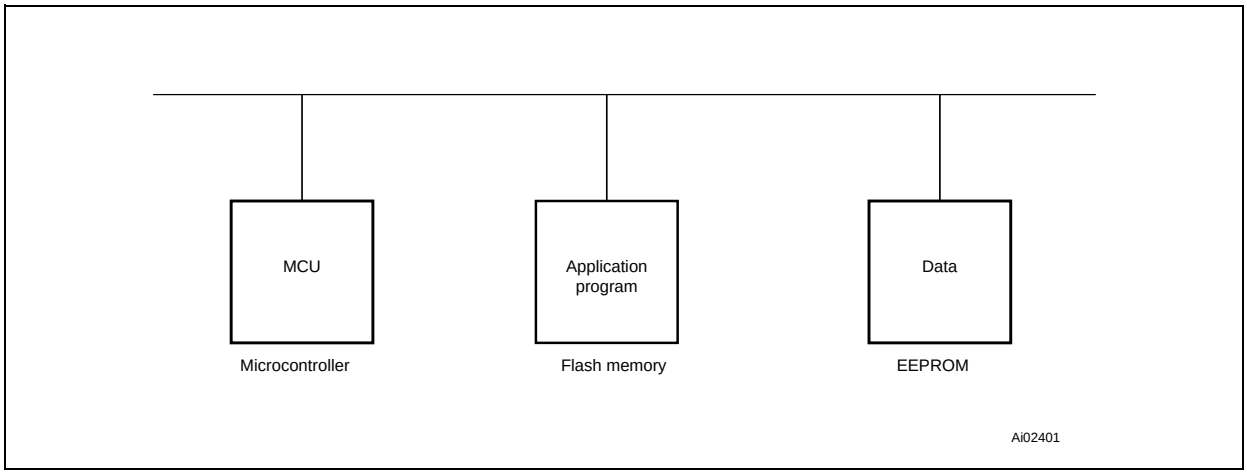
1. one microcontroller (complete with scratch memory areas)
2. one flash memory, to hold the executable code and persistent data.

This arrangement has the advantage of compactness and simplicity, but is not always an acceptable solution. Flash memory allows bytewise writing to bytes that have previously been erased, but the erase operation can only be performed in a sectorwise (or devicewise) fashion. (Sector sizes can be 8 Kbyte, or more, depending on the manufacturer).

APPLICATIONS USING A MICROCONTROLLER CHIP AND TWO EXTERNAL MEMORIES

Due to the widespread market-availability of EEPROM and flash memory, this solution, depicted in Figure 1, is the one that is frequently adopted. The main disadvantage of this approach is the need to have two memory packages, and consequently the relatively large PCB area that is occupied.

Figure 1. Application Based on Separate Flash Memory and EEPROM



COMBINING THE FUNCTIONALITY OF FLASH MEMORY AND EEPROM IN A SINGLE PACKAGE

It would be preferable not to have to include several memory chips on the PCB, but to have the functionality of each of the types of memory combined in a single package. This would, in fact, be advantageous to all microcontroller applications. For those that previously could not justify the cost of having different types of memory in the system, the increased functionality becomes available at little or no extra cost. For applications that previously needed separate integrated circuit packages for each memory type, multi-function memory either offers the opportunity to incorporate more functionality within the same area of PCB, or offers the advantages of reducing the PCB area and chip count, as summarized in the following list:

- miniaturization, due to smaller PCBs
- reduced component purchasing and assembly costs
- faster testing (or more thorough testing)
- increased system reliability
- better use of silicon area through the sharing of common functions
- increased system quality
- lower power consumption.

The appearance of having flash memory and EEPROM, fabricated together on a single chip, can be emulated by either of the following strategies:

1. by fabricating a large EEPROM device, and using a part of it as Flash memory
2. by fabricating a large Flash memory device, and using a part of it as EEPROM.

The first approach is not economically viable: the standard EEPROM process is based on a two-transistor cell, whereas Flash memory uses a single transistor memory cell (as described in Application Note, *AN998*). Since today's applications demand large Flash memory capacities, of 4 Mb or more, this two-fold bloating of the required silicon area is too high a price to pay for the increased functionality.

The second approach is more attractive, and can be undertaken in either of the following ways:

a. emulation in software of the EEPROM area (ESE)

b. emulation in hardware of the EEPROM area.

APPLICATIONS USING EMULATION IN SOFTWARE OF THE EEPROM AREA

This technique takes one large flash memory, and splits it into two functional areas:

i. one area is used as normal flash memory, for holding the executable program

ii. the remaining area (typically two sectors in size) is used for the emulation of the EEPROM.

To emulate the bytewise update capability of EEPROM in flash memory, each emulated byte is represented as a linked list within a newly erased sector of flash memory. A second sector is reserved for taking over when the first sector becomes full, and is erased in readiness. Each data byte is accompanied by a two-byte address pointer whose initial value is 'null' (the erased state). Each write operation involves adding an element on to the end of the linked-list for the byte that is being addressed.

Each read operation in the emulated EEPROM involves finding the first entry for the byte that is being addressed, and then tracing down through the linked-list until the end of the list is reached. The read operation, therefore, can involve a very long sequence of read cycles.

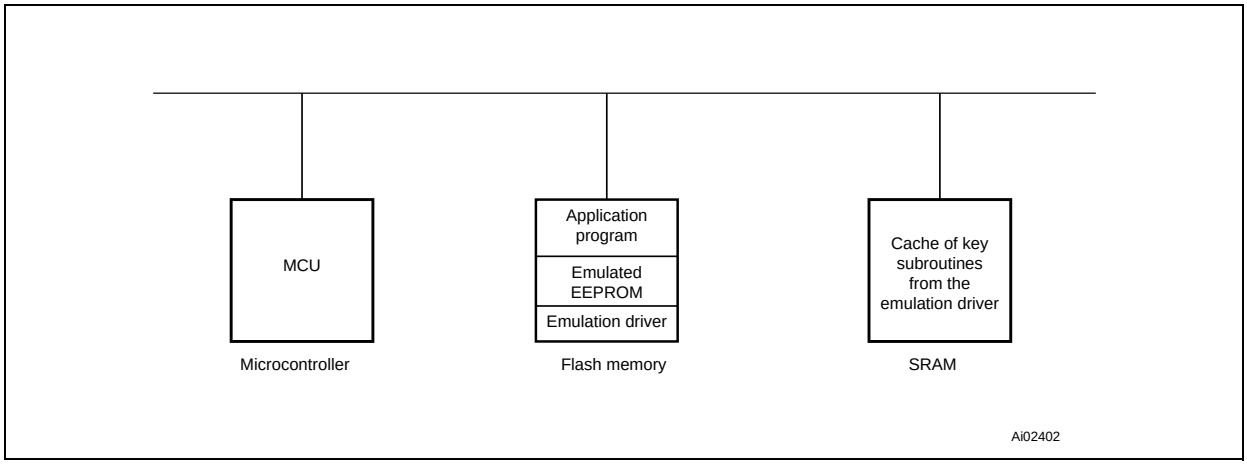
When the sector becomes full, the last element of each linked list (the most recent value of each stored byte) is copied across to the second sector, and that sector takes over as the current state of the emulated EEPROM. The first sector is erased, in readiness for taking over again, once the linked lists have grown too large in the second sector.

In applications where the data are changed frequently, the read access time of the emulated EEPROM can be hundreds of times longer than the read access time of a true EEPROM. This would be a major concern for applications in which the processor has to perform frequent data searches.

Another problem is that flash memory cannot perform simultaneous reads and writes, even to addresses in unrelated sectors. This limitation, of course, applies to all single-ported memory. However, it is particularly significant in this case: every read and write to the emulated sectors of the flash memory involves a long sequence of instruction fetches for the emulator software. These instructions, therefore, need to be stored in another memory device (usually in an external static RAM chip) if the fetching of each one is not to interfere with normal user-access of the flash memory.

Figure 2 depicts the partitioning of the various memory regions. The emulation driver generally takes up 16 Kbytes of the flash memory, but only 1 to 4 Kbytes of this needs to be copied into SRAM at any one time. The emulated EEPROM takes a further two sectors from the flash memory, leaving the remainder available for use as normal flash memory (for the application program, for example).

Figure 2. Application Based on a Single Bank of Flash Memory



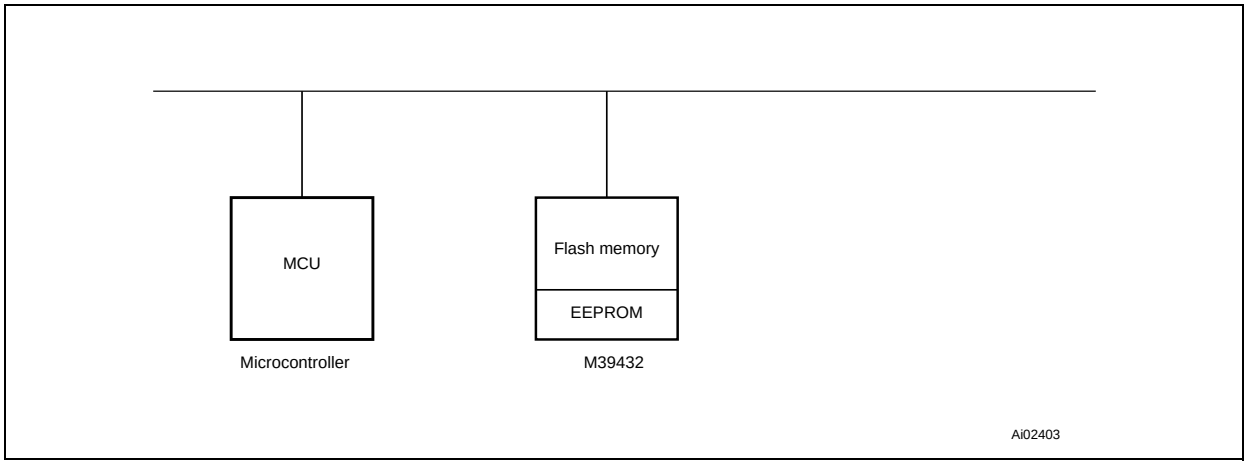
The drawbacks of the ESE approach include:

1. extra software complexity, caused by the emulation software
2. the need for a data buffer for the sector that is being changed
3. the long write time
4. all other memory operations are suspended, including instruction fetch, during the long write time
5. the need, still, for two types of memory chip to be included on the PCB.

APPLICATIONS USING EMULATION IN HARDWARE OF THE EEPROM AREA

ST's FLASH+ technology provides EEPROM functionality that is emulated using conventional Flash technology. A key benefit is that the host controller can read the Flash memory while an EEPROM write cycle is in progress; the EEPROM block manages its write cycle internally once the host controller has issued the program command.

Figure 3. Application Based on the M39432 Device



FLASH+ offers the power-consumption and footprint advantages of the single memory package solution, along with the speed advantages of the dual memory device solution. It is particularly well suited for use in the portable-applications market, as highlighted by the numbered points that follow.

1. Small footprint:

   FLASH+ brings savings in PCB size, fabrication steps and a consequent improved overall product quality. An M39432 in a single TSOP40 occupies about 40% less PCB area than a standard 4 Mb TSOP32 Flash memory, plus a 256 Kb TSOP28 parallel EEPROM.

   Within the FLASH+ device, the Flash memory and EEPROM blocks share the same data, address and control lines. The only extra overhead is the distinction between $\overline{EF}$ (Enable Flash) and $\overline{EE}$ (Enable EEPROM), and the R/$\overline{B}$ (Ready/$\overline{Busy}$) output (as depicted in the functional block diagram of Figure 4).

   The 40-pin TSOP40 arrangement (as shown in Figure 5) is forward-compatible with (that is, an extended form of) the 32-pin M29W040 Flash memory. Pins 3 to 18 and 23 to 38 correspond exactly with the pins of the M29W040 device.

2. Low power consumption:

   There is a single supply voltage, across the Vcc and Vss pins. The stand-by current can be as low as 5 µA, over the full voltage and temperature ranges, when the device is in its Deep-Power-Down mode.

Though much higher than this, the full operating current is required only during processor read and write cycles. This gives the FLASH+ device a much lower power consumption than is possible with the ESE technique: under ESE, each read or write cycle is composed of a sequence of microprocessor read and write cycles (often, several hundreds of them), the full operating current is required for longer periods than it is for the FLASH+ device.

3. High speed:

   When the application is running, the processor fetches executable instructions from the Flash memory, and data from the EEPROM, occasionally writing data to the EEPROM. The EEPROM region of the FLASH+ device can be read or written a byte at a time, or a 64-byte page at a time.

   Although writing to the EEPROM region requires a few milliseconds, the FLASH+ device is accessed as two independent memories. This *concurrent mode* of operation allows the Flash memory to continue to service read requests, such as instruction fetches, even while the EEPROM is in the middle of performing a write or erase operation.

4. Short program development time:

   Using FLASH+, the designer does not need to write EEPROM emulation software, or to fine-tune existent emulation software for the pattern of read, write, erase and interrupt requests that occur in the particular application in hand.

5. Traceability and tagging:

   The M39432 contains a 64 byte OTP row (one-time-programmable). This can be *written once*, using a dedicated multi-byte instruction (see Table 4 of the M39432 data sheet). The OTP row can be used for any user-defined purpose (such as for the holding of serial numbers, fabrication parameters, and other tags for identification and traceability). The first write operation (whether it be a byte or page write) to the OTP row will freeze the whole row against any further writes.

6. Large memory:

   The FLASH+ device is ideally suited to applications that need a large array of executable code and a medium/large array of persistent, but changeable, data.

   The two independent blocks of memory are each organized on an 8-bit bus. For the M39432, for example, one block consists of 4 Mb of Flash memory (arranged as 512 K by 8 bits), and the other of 256 Kb of parallel EEPROM (arranged as 32 K by 8 bits).

FURTHER READING

Table 3 lists the application notes that describe ST's FLASH+ technology and M39432 memory device.

Table 3. Bibliography of the Application Notes on FLASH+ Technology

Application Note	Content	Title
AN997	Introduction	M39432: a FLASH+™ Multiple Memory Device
AN931	Greater detail	On-Chip Hardware EEPROM Emulation versus Flash Memory Software Solutions
AN996	Glossary	Emulating the Unification of Flash and EEPROM: a Glossary
AN998	Technology	FLASH+™ Multiple Memory Technology
AN999	Drivers	Software Drivers for the M39432 FLASH+™ Multiple Memory

SUMMARY OF THE FEATURES OF THE M39432 FLASH+ DEVICE

- Uniform access time: 120 ns (Flash memory and EEPROM)
- Single, uniform supply voltage: 3.0 V to 3.6 V (for read, program and erase operations)
- Low stand-by current: 5 µA in deep power-down
- Automatic stand-by and deep power-down modes
- Device-type and memory-type identifier
- Write, program and erase status bits
- Data retention: 10 years
- Standard memory package: TSOP40 (10 x 20 mm)
- Extended temperature range: -40 °C to +85 °C

FLASH MEMORY BLOCK (IDENTICAL TO THE M29W040 FLASH MEMORY)

Programming time: 10 ms per byte

Flash sector erase

- Erase time: 1.5 seconds per sector
- 8 Sectors of 64 Kbytes each
- Sector protection
- Multi-sector erase
- Erase suspend and resume

10,000 Program/Erase cycles per sector

EEPROM BLOCK

- 512 pages of 64 bytes
- Additional 64 byte one-time-programmable (OTP) page

Writing features

- 64 byte page-write or single byte-write: 10 ms (maximum)
- Ready/$\overline{\text{Busy}}$ output pin
- Software data protection

100,000 write cycles

Figure 4. Functional Block Diagram

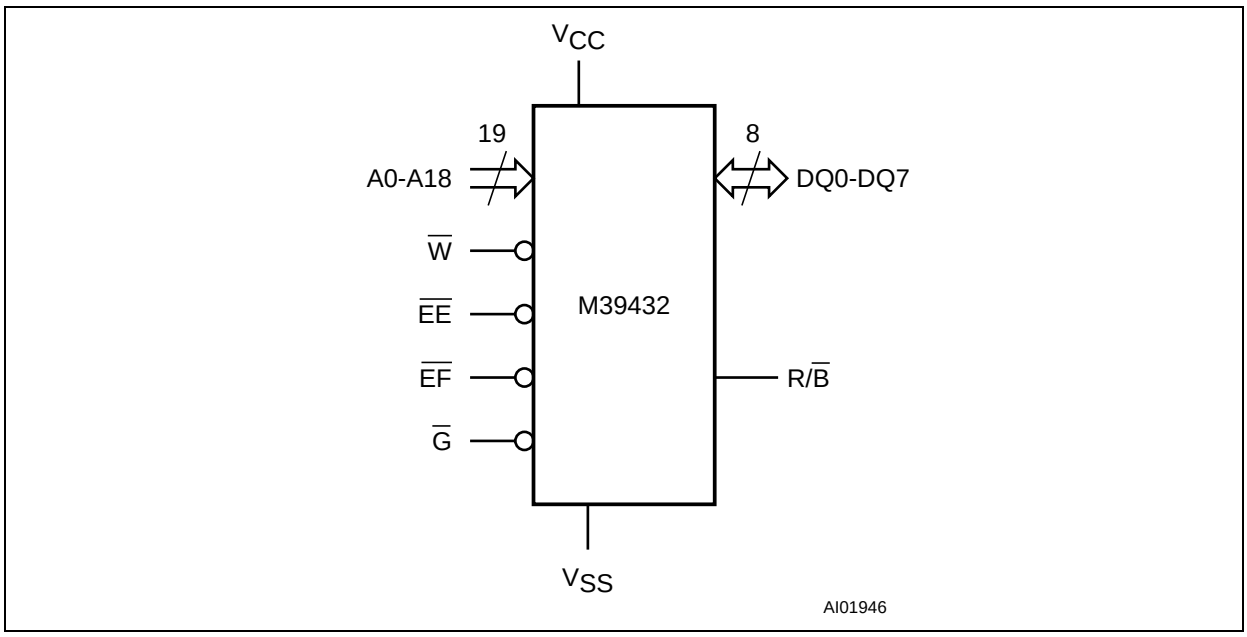
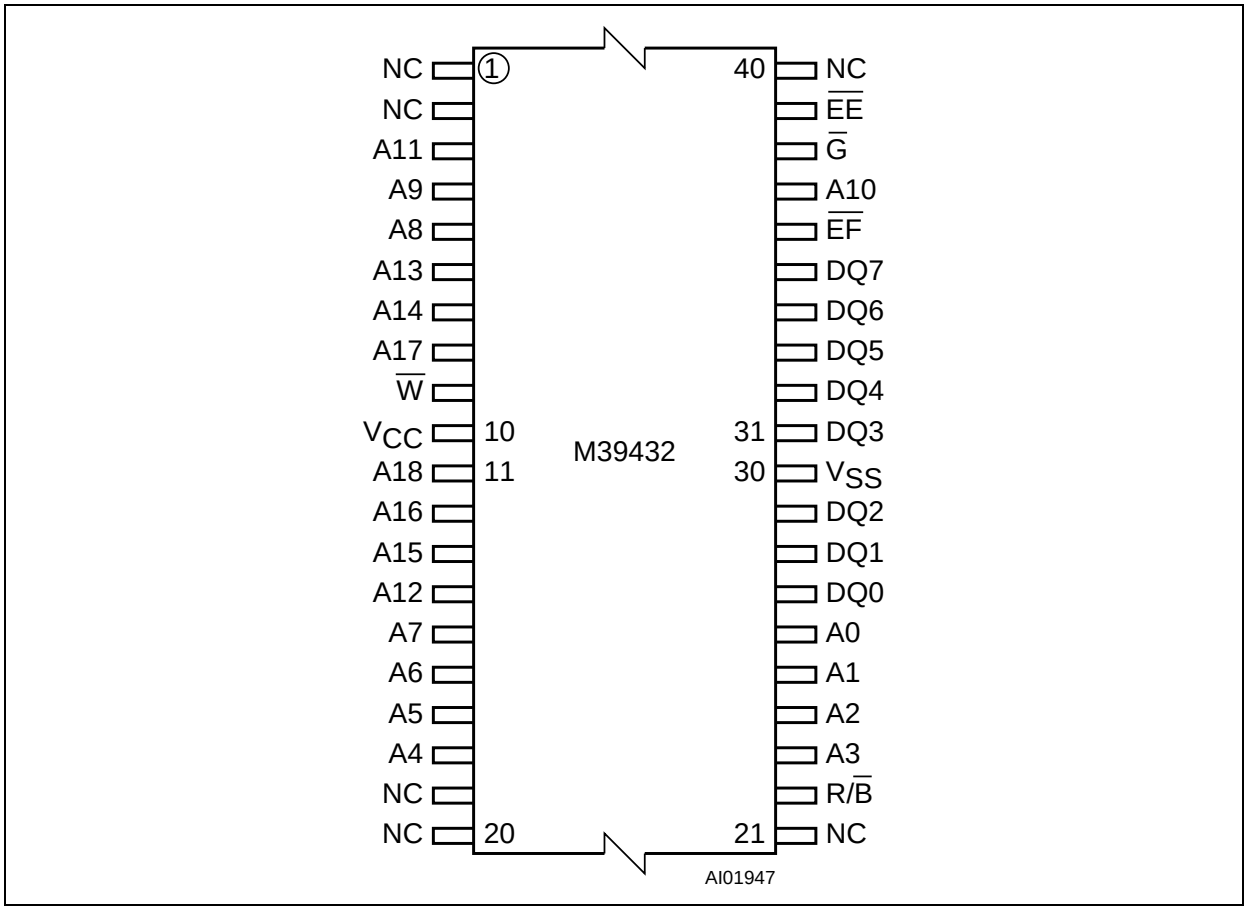


Figure 5. M39432 TSOP40 Pin-out



If you have any questions or suggestions concerning the matters raised in this document, please send them to the following electronic mail address:

apps.eeprom@st.com

Please remember to include your name, company, location, telephone number and fax number.